

UNIVERSIDAD AUTÓNOMA DE MADRID
ESCUELA POLITÉCNICA SUPERIOR



**Grado en Ingeniería de Tecnologías y Servicios de
Telecomunicación**

TRABAJO FIN DE GRADO

App Android para Exámenes de Respuestas Múltiples

Belén Gómez Hernández
Tutor: Eduardo Boemo Scalvinoni

Julio 2019

App Android para Exámenes de Respuestas Múltiples

AUTOR: Belén Gómez Hernández
TUTOR: Eduardo Boemo Scalvinoni



Digital System Laboratory

Escuela Politécnica Superior
Universidad Autónoma de Madrid
Julio de 2019

Resumen

Este Trabajo Fin de Grado consiste en desarrollar una aplicación Android para realizar *test* sobre contenidos de la asignatura de Circuitos Electrónicos Digitales impartida en el Grado de Ingeniería de Tecnologías y Servicios de Telecomunicación de la Escuela Politécnica Superior de la Universidad Autónoma de Madrid.

El objetivo principal es proporcionar un medio adicional para la verificación de conocimientos sobre la asignatura. Además se pretende proporcionar al estudiante una herramienta para practicar exámenes tipo test que se encontraran posteriormente en el aula. En total están disponibles cuatro test con ejercicios sobre Mapas Veitch-Karnaugh, Circuitos Combinacionales y Electrónica básica.

La aplicación está disponible en 149 países de forma gratuita y es accesible desde cualquier lugar ya que no necesita conexión a la red para su funcionamiento. Buscamos que cualquiera se pueda beneficiar de este proyecto por lo que la aplicación está desarrollada en inglés y está publicada en la tienda de aplicaciones de Google Play Store.

Para poder mejorarla, hemos añadido un apartado en el que el usuario pueda reportar posibles errores y sugerencias para hacer una aplicación más completa y conseguir un producto fácil, accesible y provechoso.

En esta memoria se justifica la elección del sistema operativo con el que trabajamos así como el diseño que tiene la aplicación y el desarrollo que se ha llevado a cabo. Se ha mantenido el formato de las aplicaciones desarrolladas previamente por el laboratorio de la universidad (DSLAb). Por último se ha hecho un análisis de los resultados y conclusiones que se han obtenido.

Palabras clave

Aplicación, App, Android, test, dispositivo, mapas Veitch-Karnaugh, Circuitos Combinacionales, Electrónica básica, Correo, Información.

Abstract

This Bachelor Thesis consists of developing an Android application to perform a test on the contents of the Digital Electronic Circuits course taught in the Engineering Degree in Telecommunication Technologies and Services at the Escuela Politécnica Superior of the Universidad Autónoma of Madrid.

The main objective is to provide additional support for obtaining knowledge on the subject, also aims to provide the user a tool to practice the model of multiple-choice test to be found later in the classroom. In total, four tests are available with exercises on Veitch-Karnaugh Maps, Combinational Circuits and Electronical considerations.

The application is available in 149 countries free of charge and is accessible from anywhere as it does not need a network connection to operate. We are looking for anyone to benefit from this project so the application is developed in English and is published in the Google App Store, Play Store.

In order to be able to improve it, we have added a section in which the user can report possible errors and suggestions to make a more complete application and obtain an easy, accessible and profitable product.

This report describes the choice of operating system with which we work as well as the design that has the application and the development that has been carried out, we have maintained the entity of the applications previously developed by the laboratory of the university (DSLAb). Finally, an analysis has been made of the results and conclusions that have been obtained.

Keywords

Application, App, Android, test, device, Veitch-Karnaugh maps, Combinational Circuits, Electronical considerations, E-Mail, Information.

Agradecimientos

En primer lugar quiero dar las gracias a mis padres, Bernabé y Belén por los cuales soy quien soy y hacia quienes solo puedo expresar mi sincero agradecimiento por apoyarme siempre. A mi hermana Sheila, por ser mi referencia y fuente de inspiración en la vida.

A mi compañera Isabel, por los buenos momentos que hemos vivido y por las risas que nos han dado fuerzas para conseguir todo lo que nos hemos propuesto.

A mi tutor Eduardo, por darme la oportunidad de realizar este TFG.

Gracias a todos aquellos que forman parte de mi vida y que me hacen crecer hacia algo mejor.

ÍNDICE DE CONTENIDOS

1 Introducción	1
1.1 Motivación.....	1
1.2 Objetivos.....	1
1.3 Organización de la memoria.....	2
2 Estado del arte	3
2.1 Sistemas operativos	3
2.1.1 iOS.....	3
2.1.2 Android.....	4
2.1.2.1 Historia	4
2.1.2.2 Características.....	4
2.1.2.3 Arquitectura	5
2.1.2.4 Versiones	6
2.2 Aplicaciones similares.....	7
3 Diseño.....	9
3.1.2.1 Botones	9
3.1.2.2 Colores.....	9
3.1.2.3 Letra.....	10
4 Desarrollo.....	15
4.1 Inicio aplicación	15
4.2 Herramientas utilizadas	15
4.3 Elementos básicos de una aplicación en Android.....	16
4.3.1 Activity	16
4.3.2 Views	17
4.3.3 Manifest	17
4.4 Bloques aplicación.....	19
4.5 Diseño y desarrollo	19
4.5.1 Test	19
4.5.1.1 Ampliar imagen	20
4.5.1.2 Persistencia de datos	21
4.5.1.3 Respuesta correcta	23
4.5.1.4 Calcular nota.....	24
4.5.1.5 Test aleatorio	26
4.5.2 Información.....	26
4.5.3 Correo	27
4.6 Adaptación pantallas.....	28
4.6.1 ScrollView	28
4.6.2 Wrap_content y Match_parent	29

5 Integración, pruebas y resultados	31
5.1 Publicar App	31
5.1.1 Generar fichero APK	31
5.1.2 Cuenta desarrollador en Google Play	33
5.1.3 Publicación en Google Play	34
5.2 Análisis de la aplicación	35
6 Conclusiones y trabajo futuro.....	36
6.1 Conclusiones.....	36
6.2 Trabajo futuro	36
Referencias	39
Glosario	40
Anexos.....	I
A Manual de instalación	I
B Manual de usuario	III
C Preguntas test.....	V

ÍNDICE DE FIGURAS

FIGURA 1. LOGO ANDROID.....	4
FIGURA 2. PILA DE SOFTWARE DE ANDROID.....	5
FIGURA 3. VERSIONES ANDROID.....	6
FIGURA 4. MENÚ COMB 2.....	10
FIGURA 5. SPLASH COMB 2.....	11
FIGURA 6. PREGUNTA Y RESULTADO TEST COMB 2	12
FIGURA 7. INFORMACIÓN COMB 2	12
FIGURA 8. CORREO COMB 2	13
FIGURA 9. CICLO DE VIDA DE UNA ACTIVIDAD.....	16
FIGURA 10. BLOQUES COMB 2.....	19
FIGURA 11. ESTRUCTURA PREGUNTA TEST	20
FIGURA 12. ARCHIVO BUILD.GRADLE	21
FIGURA 13. TABLA RESULTADOS TEST COMB 2	26
FIGURA 14. ESTRUCTURA DISEÑO PREGUNTA TEST	28
FIGURA 15. VISTA DISEÑO PREGUNTA TEST.....	28
FIGURA 16. PRIMER PASO GENERAR FICHERO APK.....	31
FIGURA 17. SEGUNDO PASO GENERAR FICHERO APK.....	32
FIGURA 18. TERCER PASO GENERAR FICHERO APK.....	32
FIGURA 19. CUARTO PASO GENERAR FICHERO APK (FIRMA).....	33
FIGURA 20. CLASIFICACIÓN APLICADA A COMB 2	34
FIGURA 21. CÓDIGO QR COMB 2.....	I
FIGURA 22. PLAY STORE COMB 2.....	II
FIGURA 23. INICIO COMB 2.....	III

ÍNDICE DE TABLAS

TABLA 1. CUOTA MERCADO SO	3
TABLA 2. APPS PLAY STORE	7
TABLA 3. APPS DSLAB.....	8
TABLA 4. PREGUNTAS TEST.....	XVI

1 Introducción

1.1 Motivación

El uso de la tecnología móvil se ha extendido exponencialmente en la última década, se han producido cambios en muchos campos y uno de ellos es en la educación. Cada vez es más habitual que los alumnos se ayuden de sus dispositivos móviles, tabletas u ordenadores personales en las aulas de nuestra universidad. Por ello, hemos creído conveniente adaptarnos a las necesidades de los alumnos y ayudarlos en su progreso ajustando los métodos docentes a sus entornos más utilizados. Además, el uso de dispositivos móviles ofrece ciertas ventajas:

- Aprendizaje dinámico y autónomo.
- Reemplazar el papel por dispositivos tecnológicos.
- Comprobar los resultados de los ejercicios en el momento.
- Libertad de desplazamiento (solo necesitamos un dispositivo).
- No es necesario conexión con la red.

La idea es desarrollar una herramienta que simule el momento en el que el alumno debe enfrentarse a un examen tipo test en una asignatura de electrónica. A parte del estudio previo que realiza el alumno, esto le garantiza que los resultados sean verídicos y contrastados en tiempo real ya que podrá saber la nota que obtendría en el test con esta aplicación que simula preguntas similares a las que el alumno tendrá que hacer frente en un examen de la asignatura.

1.2 Objetivos

El objetivo principal de esta aplicación es ofrecer una herramienta de ayuda a los alumnos de la asignatura de Circuitos Electrónicos Digitales de la EPS, aunque se ha desarrollado para ser de utilidad a cualquier persona que lo desee (está implementada en inglés y se ha puesto a disposición en 149 países de forma gratuita).

Esta aplicación se basa en los exámenes tipo test que los alumnos realizan en la asignatura, por lo que les será útil para estudiar y repasar los contenidos que se traten en el aula.

Todos los estudiantes que dispongan de un dispositivo compatible con el sistema operativo Android (el más utilizado globalmente), podrán instalarse la aplicación gratuitamente desde la plataforma de Google Play Store.

Por otra parte, el objetivo buscado para la persona que desarrolla este TFG es aprender a implementar aplicaciones en Android e introducirse en la programación con el lenguaje Java. Estas materias no quedan cubiertas por las asignaturas obligatorias del Perfil en Diseño e Implementación de Sistemas Electrónicos de Comunicaciones, por lo que este trabajo será de ayuda para completar la formación obtenida.

1.3 Organización de la memoria

Esta memoria consta de los siguientes apartados:

- Capítulo 1: Introducción, objetivos y organización de la memoria.
- Capítulo 2: Estado del arte y aplicaciones en el mercado.
- Capítulo 3: Diseño, requisitos y bloques de la aplicación.
- Capítulo 4: Desarrollo de la aplicación.
- Capítulo 5: Integración, pruebas y análisis.
- Capítulo 6: Conclusiones y trabajo futuro.

Anexo A: Manual de instalación.

Anexo B: Manual de usuario.

Anexo C: Preguntas test.

2 Estado del arte

En este apartado analizamos los sistemas que existen para poder realizar un proyecto similar al que estamos desarrollando en este trabajo.

2.1 Sistemas operativos

Nos referimos como sistema operativo de un dispositivo al conjunto de programas que hace posible la empleabilidad del hardware de este. En el ámbito de la telefonía móvil nos encontramos con diversos SO como Android, iOS, Windows Phone, BlackBerry, etc. pero sin duda los sistemas punteros en este sector son iOS y Android que abarcan el mayor porcentaje en la cuota de mercado.

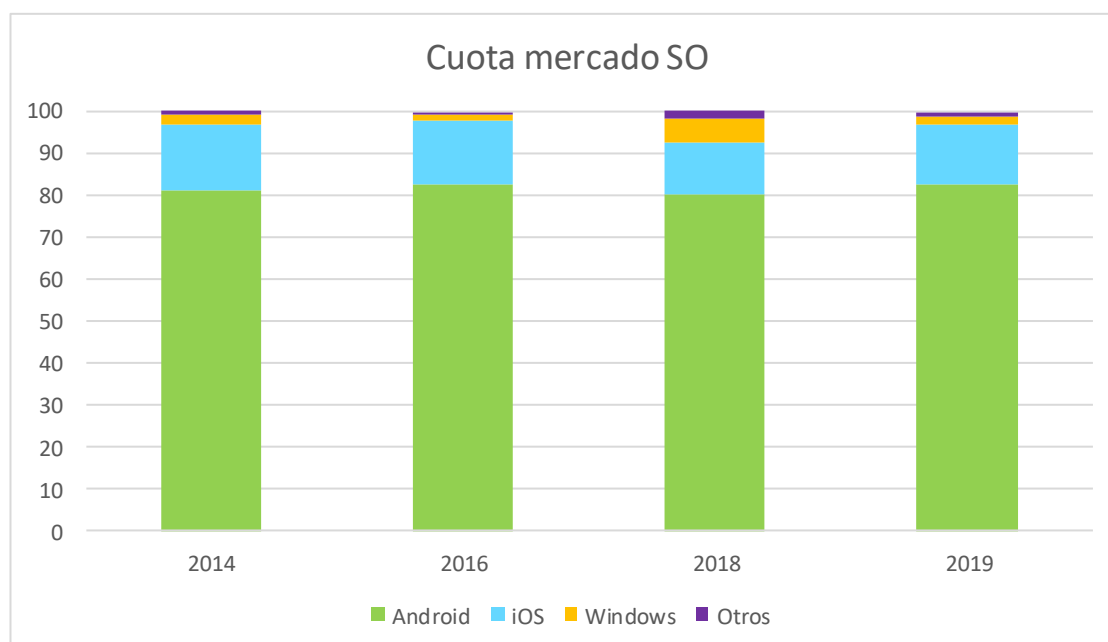


Tabla 1. Cuota mercado SO

2.1.1 iOS



Este SO fue dado a conocer en 2007, está basado en código cerrado desarrollado por Apple Inc. y no fue hasta un año después de su lanzamiento que el SDK fue liberado para el uso de terceros desarrolladores.

No permite la instalación libre, de forma que solo los dispositivos de la marca pueden implantar este sistema. Es el segundo SO más demandado hoy en día, aunque compite muy por debajo del imponente sistema operativo Android.

Figura 1. Logo iOS

2.1.2 Android

2.1.2.1 Historia



Android es un sistema operativo desarrollado por Android Inc. que en 2005 fue comprada por la multinacional Google, en 2008 lanzó su primera versión del sistema basado en el Kernel de Linux. Se trata de un sistema operativo de código abierto, lo que permite a cualquier desarrollador crear aplicaciones en las condiciones de programación que desee, además permite su implantación en cualquier terminal compatible.

Cuenta con el mayor apoyo de los usuarios liderando la mayor cuota de mercado de SO de telefonía móvil de la historia.

Figura 1. Logo Android

2.1.2.2 Características

Las características más destacables de Android son:

Máquina virtual Dalvik: Android utiliza su propia máquina virtual dejando atrás el uso de la tradicional máquina virtual Java (VM), esto asegura que se ejecute de manera eficiente en un dispositivo la multitarea. La multitarea de Android hace que cuando no se estén ejecutando aplicaciones en primer plano, reciban ciclos de reloj.

Almacenamiento de datos SQLite: SQLite es un software libre de configuración sencilla que almacena los datos de forma que aunque se apague el dispositivo los datos persistan y se encuentren en la aplicación cuando lo encendamos.

Navegador basado en WebKit: cuenta con herramientas como la opción de navegación privada o la barra unificada de URL y búsqueda entre otras muchas cosas.

Conectividad y mensajería: Android soporta diversas tecnologías de conectividad como Bluetooth, Wi-fi, GPRS, etc., además de mensajería SMS y MMS.

Multimedia: tiene soporte para múltiples formatos multimedia de audio, video e imágenes. También soporta el hardware asociado como cámaras de fotos, pantallas táctiles, GPS, sensores, etc.

2.1.2.3 Arquitectura

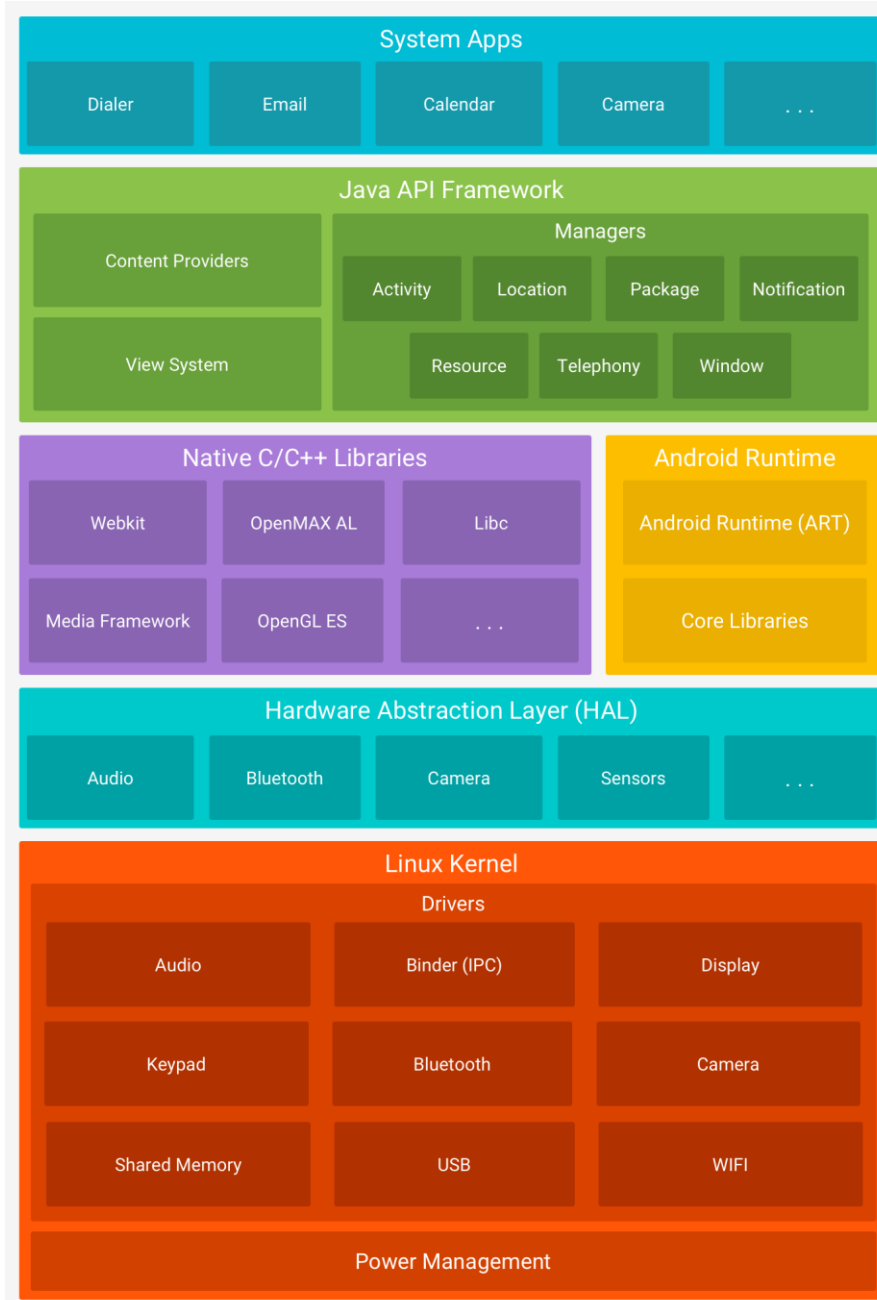


Figura 2. Pila de software de Android

La arquitectura interna de Android se basa en las siguientes capas de software libre:

Kernel de Linux: permite a Android aprovechar las funciones de seguridad clave al mismo tiempo que permite el desarrollo controlado de hardware para un kernel conocido.

Capa de abstracción de hardware (HAL): ofrece interfaces estándares para las capacidades hardware del dispositivo.

Tiempo de ejecución de Android: para dispositivos con versiones de nivel API 21 o posteriores, cada app ejecuta procesos individuales con sus instancias de tiempo de ejecución de Android (ART).

Bibliotecas C/C++ nativas: muchos servicios de Android como el ART o el HAL, se basan en código nativo que requiere bibliotecas nativas escritas en C y C++.

Framework de la Java API: todas las funciones de Android están disponibles mediante API escritas en Java. Esto es la base para crear apps reutilizando los componentes del sistema y servicios centrales. Los desarrolladores tienen libre acceso a las API del framework que usan las apps de Android.

Apps del sistema: Android incluye apps centrales para correo electrónico, mensajería, calendario, navegación en internet, contactos, etc. Estas aplicaciones funcionan como apps para los usuarios y los desarrolladores pueden acceder a ellas por medio de sus propias apps.

2.1.2.4 Versiones

Desde que se lanzara la primera versión en 2008 (“Apple Pie”), Android no ha parado de desarrollarse y mejorar, se han creado 27 niveles de API. Estas actualizaciones buscan corregir errores y agregar nuevas funcionalidades. Las nuevas versiones que se lanzan son compatibles con las anteriores, aunque por el contrario, para versiones posteriores podemos encontrarnos con incompatibilidades en los dispositivos no actualizados.

En la siguiente tabla podemos observar las versiones más nuevas y distribución global.

ANDROID PLATFORM VERSION	API LEVEL	CUMULATIVE DISTRIBUTION
4.0 Ice Cream Sandwich	15	
4.1 Jelly Bean	16	99,6%
4.2 Jelly Bean	17	98,1%
4.3 Jelly Bean	18	95,9%
4.4 KitKat	19	95,3%
5.0 Lollipop	21	85,0%
5.1 Lollipop	22	80,2%
6.0 Marshmallow	23	62,6%
7.0 Nougat	24	37,1%
7.1 Nougat	25	14,2%
8.0 Oreo	26	6,0%
8.1 Oreo	27	1,1%

Figura 3. Versiones Android.

2.2 Aplicaciones similares

Antes de comenzar un proyecto de una aplicación, debemos comprobar lo que actualmente ofrece la plataforma en la que vamos a trabajar. En nuestro caso, evaluamos las aplicaciones que están disponibles para Android ya que es el entorno en el que competiremos.

Analizamos las Apps que presentan contenidos similares a la nuestra. En la siguiente tabla podemos comparar tres gratuitas y bien valoradas por los usuarios.




Nombre	Logo	Idioma	Usabilidad
Ingeniería Electrónica		Inglés	Tiene un exceso de publicidad. Mala implementación en los test. No permite retroceder en las preguntas. Preguntas de test sin imágenes. Salida de la App difícil para el usuario.
Ingeniería Electrónica Básica		Inglés	Copia el mismo formato que la App anterior (<i>"Ingeniería Electrónica"</i>). Aplicación simple de test.
Aprende Electrónica		Inglés Español Portugués	El usuario puede introducir preguntas con imágenes. Preguntas sin contrastar y con errores. Formato bastante complejo.

Tabla 2. Apps Play Store

Las tres aplicaciones que acabamos de conocer nos ofrecen mejorar nuestros conocimientos de Ingeniería electrónica con un temario bastante amplio, si nos centramos en el temario que se trata en la asignatura de “Circuitos Electrónicos Digitales” encontramos aplicaciones desarrolladas por alumnos de la EPS a lo largo de su historia. Estas aplicaciones tratan los mismos contenidos que queremos ofrecer a los estudiantes, por lo que también analizamos la forma en la que se presentan.

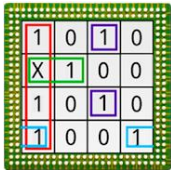

Nombre	Logo	Idioma	Usabilidad
Karn Map		Inglés	Aplicación exclusiva sobre mapas Veitch-Karnaugh. Se centra en la resolución de problemas.
Combinational Circuits		Inglés	Contiene ejercicios de multiplexores, puertas lógicas, BCD 7-segmentos, Karnaugh, codificadores y decodificadores.

Tabla 3. Apps DSLab

3 Diseño

3.1 Requisitos

Este proyecto está enfocado a reforzar contenidos de la asignatura de Circuitos Electrónicos Digitales, por tanto, al formar parte de un grupo de trabajo debemos amoldarnos al resto de aplicaciones diseñadas para la misma asignatura, buscamos que todas las aplicaciones con las que el alumno se vaya a enfrentar tengan un diseño similar con la idea de facilitarles el trabajo.

3.1.1 Sistema Operativo

Como hemos mencionado en el Estado del arte, el sistema operativo por el que hemos optado es Android ya que conforme a la gráfica de estudio de mercado vista anteriormente, es el sistema que abarca el mayor porcentaje de cuota de mercado en sistemas operativos. Además, es el sistema para el que tenemos el material necesario para el desarrollo de una nueva aplicación.

3.1.2 Interfaz gráfica

3.1.2.1 Botones

Nuestra aplicación está diseñada con dos tipos de botones: botones con imágenes y con letras.

Los botones con imágenes o `ImageButton` los hemos utilizado para distinguir aquellos botones que llevan al menú principal (en las preguntas y en el correo) y los botones de información y correo (en el menú principal). Estos botones dan una estética más dinámica a la aplicación.

Los botones con letras o `Button` se han reservado para indicar el contenido de los test en el menú principal y para hacer el enlace dentro de las preguntas con su pregunta anterior y posterior.

3.1.2.2 Colores

Nuestro objetivo es conseguir una aplicación funcional, intuitiva y eficaz por lo que para evitar un exceso colores, hemos reducido la gama de colores a blanco, negro, gris y azul.

El fondo de nuestra app siempre es blanco o similar puesto que queremos mostrar claridad en el contenido y dar iluminación a nuestra pantalla.

Las figuras son fundamentales y contienen colores básicos para distinguir los elementos que las componen.

3.1.2.3 Letra

Se ha optado por un tipo de letra con una tipografía clara y legible, además el tamaño es de mínimo 18 sp para favorecer la lectura en cualquier dispositivo.

3.1.2 Idioma

Aunque la asignatura se imparte en castellano, escogemos el inglés como idioma para nuestra aplicación ya que nuestro propósito es que sirva de utilidad a la mayor cantidad de gente posible y puesto que el inglés es la lengua vehicular elegimos esta. Además la aplicación va a ser publicada en un medio internacional y va a dar servicio a 149 países.

3.2 Bloques aplicación

La aplicación consta de un menú principal el cual tiene los enlaces para acceder a: Información, Correo y cuatro test de 15 preguntas cada uno.

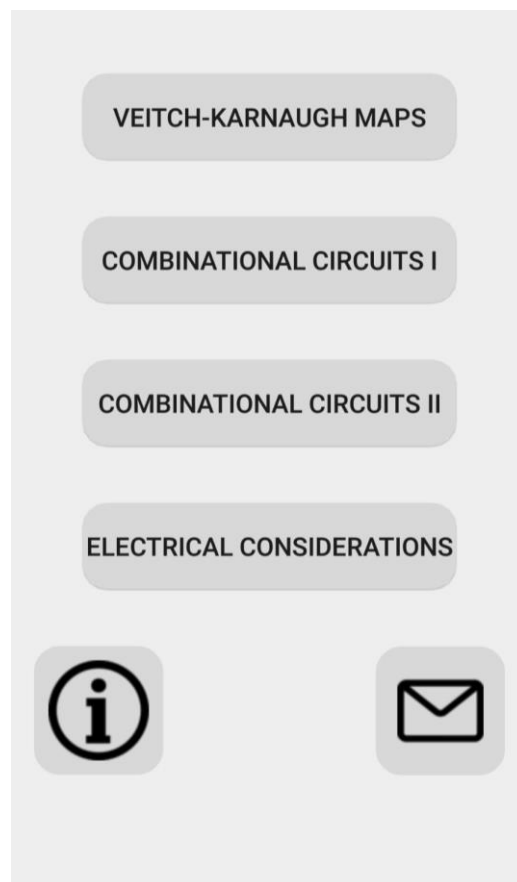


Figura 4. Menú COMB 2

3.2.1 Presentación

La aplicación dispone de un Splash (Pantalla de Inicio), dónde brevemente se expone lo que se va a encontrar en la aplicación, en este caso nuestro Splash es:

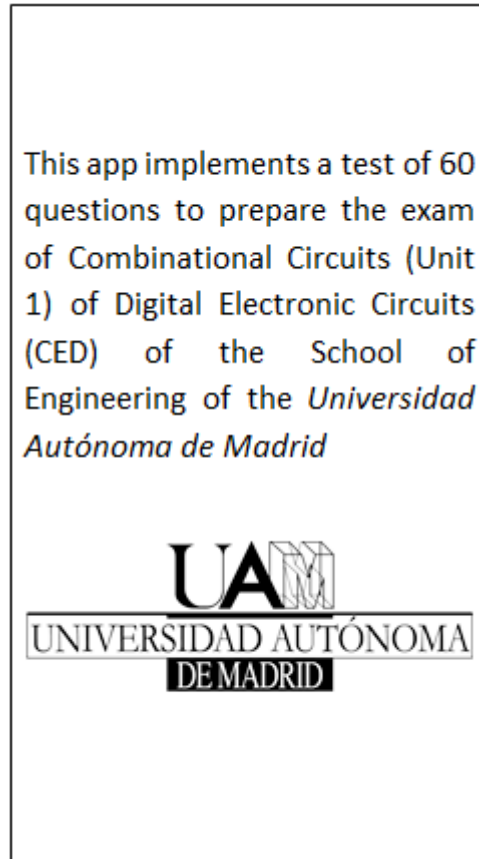
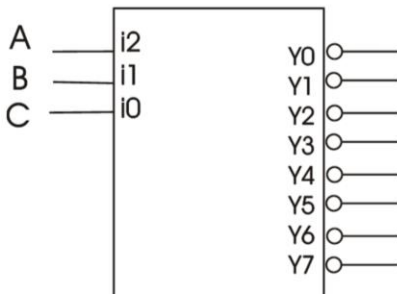


Figura 5. Splash COMB 2


3.2.2 Test

Hemos desarrollado 4 test con ejercicios sobre mapas de Veitch- Karnaugh, circuitos combinacionales y electrónica básica. Cada test consta de 15 preguntas, cada pregunta tiene 4 posibles respuestas donde solo una es verdadera (la app nunca revelará la respuesta correcta), la estructura de la pregunta está diseñada con una pregunta, una figura, las respuestas y tres botones: para ir a la pregunta anterior, avanzar a la siguiente o volver al menú principal. Al finalizar el test aparece en pantalla un cuadro con los resultados (preguntas correctas, incorrectas, sin responder y nota) y dos botones (para volver a repetir el test o volver al menú principal). Cuando se repite un test las respuestas salen en un orden aleatorio con el propósito de que el alumno razone las respuestas y no se las aprenda de memoria.

In the 3-8 decoder with active-low outputs:



☐ Y7 = 0 if ABC = 111.
☐ Y7 = 1 if ABC = 111.
☐ Y4 = 0 if CBA = 100.
☐ Y4 = 1 if CBA = 100

PREVIOUS  NEXT

The function $F[DCBA]$ is:

BA \ DC	00	01	11	10
00	1	0	0	1
01	0	1	1	0

Results
 Correct: 11
 Incorrect: 2
 Not selected: 2
 Note: 6,89

START OVER FINISH

☐ $F = / [C \text{ XOR } A]$
☐ $F = C \text{ XOR } A$


PREVIOUS  FINISH

Figura 6. Pregunta y resultado test COMB 2

3.2.2 Información

En este layout (Pantalla), encontramos la siguiente información acerca de la aplicación:



Texts and Figures: Eduardo Boemo
 Programming: Belén Gómez Hernández

Suggestions and comments are welcome at:

eduardo.boemo@uam.es
 belen.gomezh@estudiante.uam.es



Figura 7. Información COMB 2

3.2.4 Correo

Hemos habilitado un apartado en el que se puedan enviar dudas sobre la app, correcciones, mejoras, etc. con el objetivo de que con la ayuda de los usuarios obtengamos un producto más completo y de mejor calidad.

El usuario tan solo debe añadir un pequeño asunto y el cuerpo del mensaje, ya que está programado para que no sea necesario introducir ninguna dirección de correo, con esto nos aseguramos de que el correo se envía a la dirección oportuna y facilitamos esta tarea al usuario.

If you detect errors, grammar mistakes, and bugs..

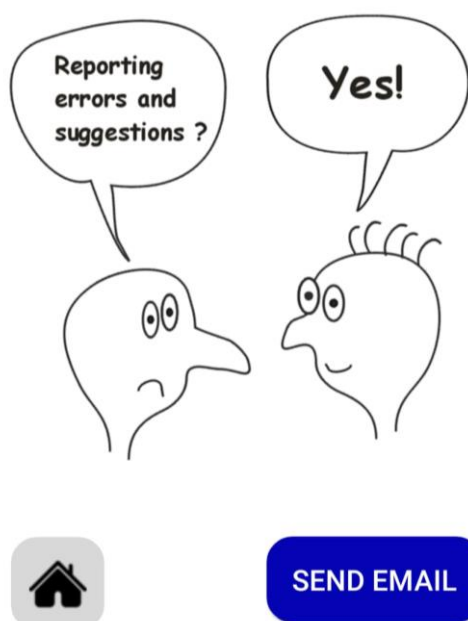
A screenshot of a web form titled 'Correo COMB 2'. The form has a light grey background. At the top, it lists 'TO: belen.gomezh@estudiante.uam.es' and 'CC: eduardo.boemo@uam.es'. Below these are two input fields: 'Subject' and 'Text', both with placeholder text 'enter subject here...' and 'enter message here...' respectively. A blue 'SEND' button is located at the bottom right. At the bottom left, there is a grey button with a black house icon.

Figura 8. Correo COMB 2

4 Desarrollo

En este apartado vamos a profundizar los pasos a seguir para conseguir el producto final, además también quedará reflejado las mejoras que hemos ido adaptando según hemos creído conveniente.

Debemos recordar que todas las acciones que realiza la aplicación conllevan un trabajo y estudio de contenido y forma para adaptar todo al dispositivo con el que se va a manejar la aplicación.

4.1 *Inicio aplicación*

Antes de comenzar el proyecto, debemos conocer las herramientas con las que se va a trabajar, y realizar estudios previos sobre los programas que vamos a necesitar controlar y los lenguajes de programación que necesitamos saber.

La programación de aplicaciones en Android tiene un nivel de dificultad que viene determinado por la experiencia del programador, ya que si estamos acostumbrados al entorno de trabajo resulta bastante intuitivo y rápido conseguir resultados. En este caso y al tener como objetivo del propio trabajo el aprender a manejar las herramientas que se necesitan, el tiempo que debemos dedicar es mucho mayor.

Como no contamos con los conocimientos necesarios para realizar una aplicación, comenzamos estudiando el lenguaje de programación requerido, en este caso, Java. Se usa Java ya que es el lenguaje con el que se desarrollan todas las aplicaciones del SDK de Android. El SDK (Software Development Kit) de Android es el kit de desarrollo de software para ejecutar aplicaciones y emular el sistema Android de cualquier versión.

4.2 *Herramientas utilizadas*

El material que vamos a necesitar para poder desarrollar este proyecto es:

- Ordenador
- Teléfono móvil que tenga el sistema operativo Android
- Android Studio
- Java SDK
- Aplicación de diseño gráfico

4.3 Elementos básicos de una aplicación en Android

4.3.1 Activity

Un activity (actividad) es un componente que contiene una pantalla con la que se puede interactuar, y tiene asignada una ventana en la que aparece su interfaz de usuario.

Una aplicación suele estar formada por varias actividades que están conectadas entre sí.

Una actividad puede existir en tres estados:

- Reanudada: La actividad está en pantalla.
- Pausada: Otra actividad está en pantalla y es visible por encima (esta actividad conserva en memoria toda la información y continua anexo al administrador de ventanas).
- Detenida: La actividad está en segundo plano (este estado conserva en memoria toda la información, pero no está anexo al administrador de ventanas). No permanece visible para el usuario y el sistema es el encargado de gestionar su eliminación.

Cuando se detiene una actividad porque iniciamos otra se produce un cambio de estado, esto viene determinado por los métodos callback del ciclo de vida de una actividad. Los métodos callback son:

- onCreate: La actividad se está creando.
- onStart: La actividad está a punto de hacerse visible.
- onResume: La actividad se ha hecho visible.
- onPause: Otra actividad va a tomar el primer plano.
- onStop: La actividad ya no es visible.
- onDestroy: La actividad se va a destruir.

El orden de los métodos sigue el siguiente esquema donde podemos diferenciar los tres estados mencionados anteriormente:

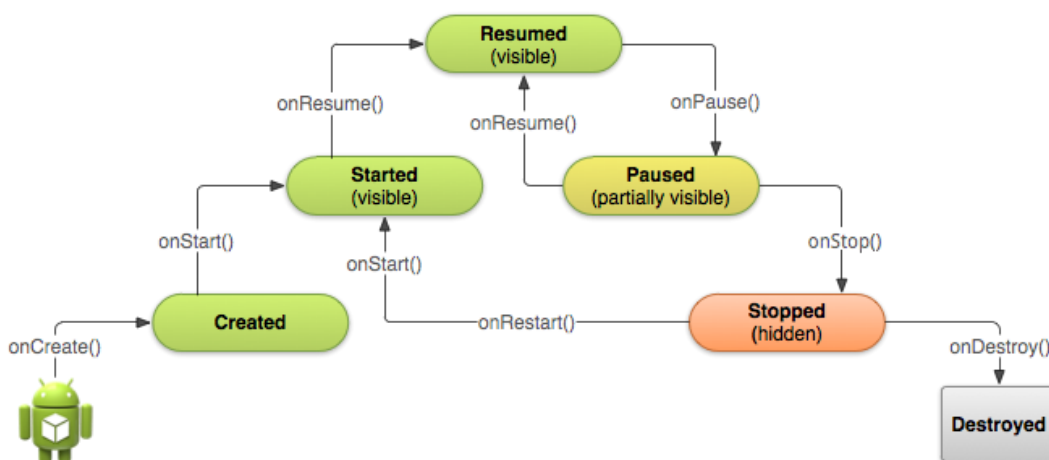


Figura 9. Ciclo de vida de una Actividad

4.3.2 Views

Esta clase representa el componente básico ya que muestra todo lo que se ve por pantalla, establece un área rectangular y gestiona el manejo de componentes interactivos como botones, imágenes, texto, etc.

Se suele diseñar en un fichero en lenguaje XML que se encuentra dentro de la carpeta “layout” de nuestra aplicación. Otra forma de implementar views es directamente en el fichero java haciendo uso de “View”.

Existen dos clases fundamentales de views:

- View (vistas): también se conoce como widgets, y representan un bloque básico, los más básicos son: Button (botones) y TextView (texto).
- ViewGroup (Contenedores): también conocido como layouts, representan conjuntos de Views. En nuestro proyecto los más utilizados son: LinearLayout y RelativeLayout.

4.3.3 Manifest

Todas las aplicaciones tienen un archivo directorio raíz, el manifiesto de la app (AndroidManifest.xml) es un archivo que proporciona información de la aplicación al sistema Android, esta información es la que el sistema debe tener para poder ejecutar el código de la app.

El archivo manifiesto:

- Nombra el paquete de Java que sirve como identificador único para la aplicación.
- Describe todo lo que integra la aplicación (componentes, actividades, etc.) y nombra las clases que implementa cada componente.
- Declara el nivel de API mínimo que debe tener la aplicación.
- Determina los procesos que establecen los componentes de la aplicación.
- Declara los permisos que debe tener la aplicación para interactuar con los componentes de la misma y para acceder a las partes protegidas de una API.

- AndroidManifest.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="es.uam.belengomez.comb2">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="COMB 2"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/AppTheme"
        tools:ignore="GoogleAppIndexingWarning">

        <activity android:name=".InicioCorreo" />
        <activity android:name=".InfoActivity" />
        <activity android:name=".CorreoActivity" />
        <activity
            android:name=".CuartoActivity"
            android:label="CuartoActivity"
            android:theme="@style/AppTheme.NoActionBar" />
        <activity
            android:name=".TerceroActivity"
            android:label="@string/title_activity_tercero"
            android:theme="@style/AppTheme.NoActionBar" />
        <activity
            android:name=".SegundoActivity"
            android:label="SegundoActivity"
            android:theme="@style/AppTheme.NoActionBar" />
        <activity
            android:name=".PrimeroActivity"
            android:label="PrimeroActivity"
            android:theme="@style/AppTheme.NoActionBar" />

        <activity
            android:name=".MainActivity"
            android:label="COMB 2"
            android:theme="@style/AppTheme.NoActionBar"/>
        <activity android:name=".SplashActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

4.4 Bloques aplicación

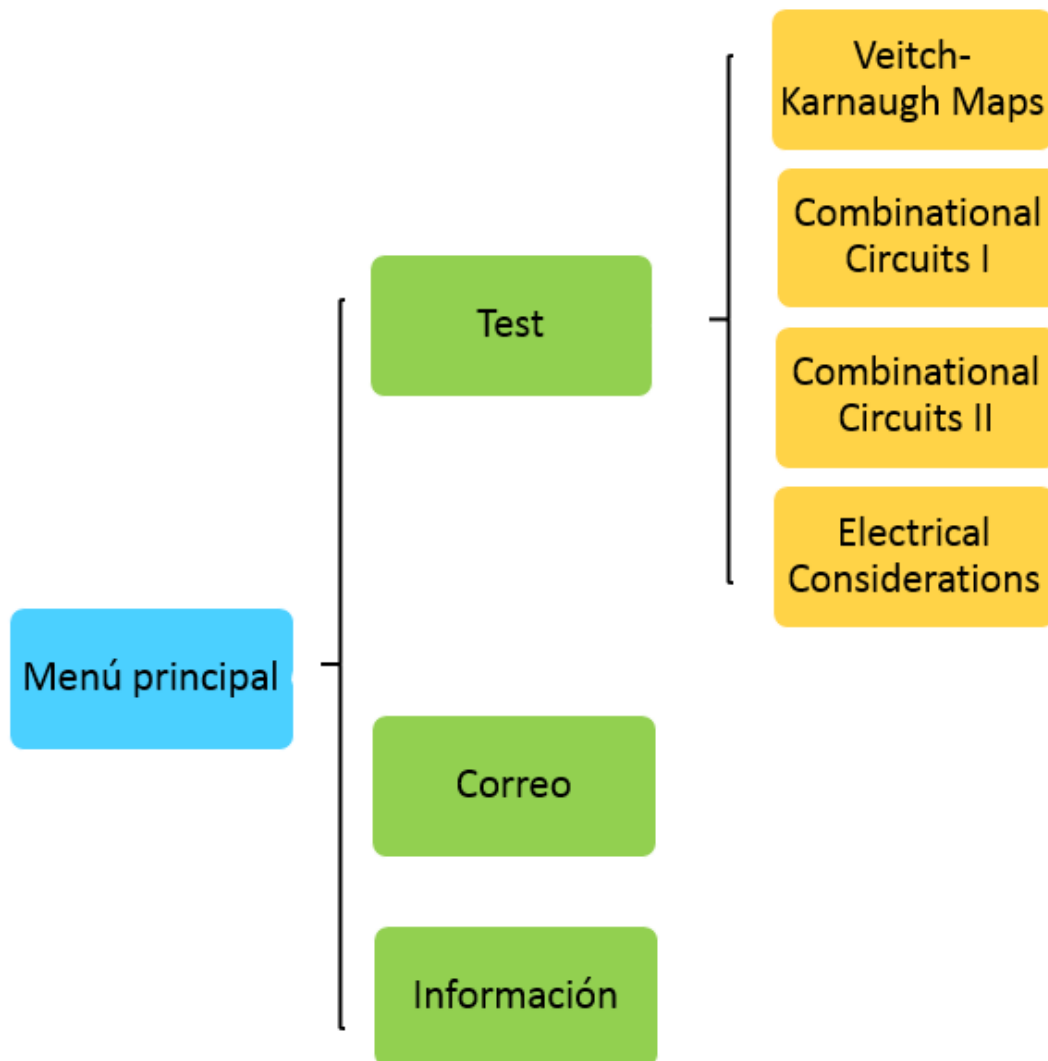


Figura 10. Bloques COMB 2

4.5 Diseño y desarrollo

4.5.1 Test

Hemos creado 4 test con ejercicios sobre la asignatura Circuitos Electrónicos Digitales. Cada test consta de 15 preguntas, el número de preguntas está fijado en 15 ya que los exámenes que se realizan en el aula son de esa extensión y buscamos ayudar al alumno en el estudio de sus capacidades para hacer frente a la asignatura.

Todas las preguntas siguen la siguiente estructura:

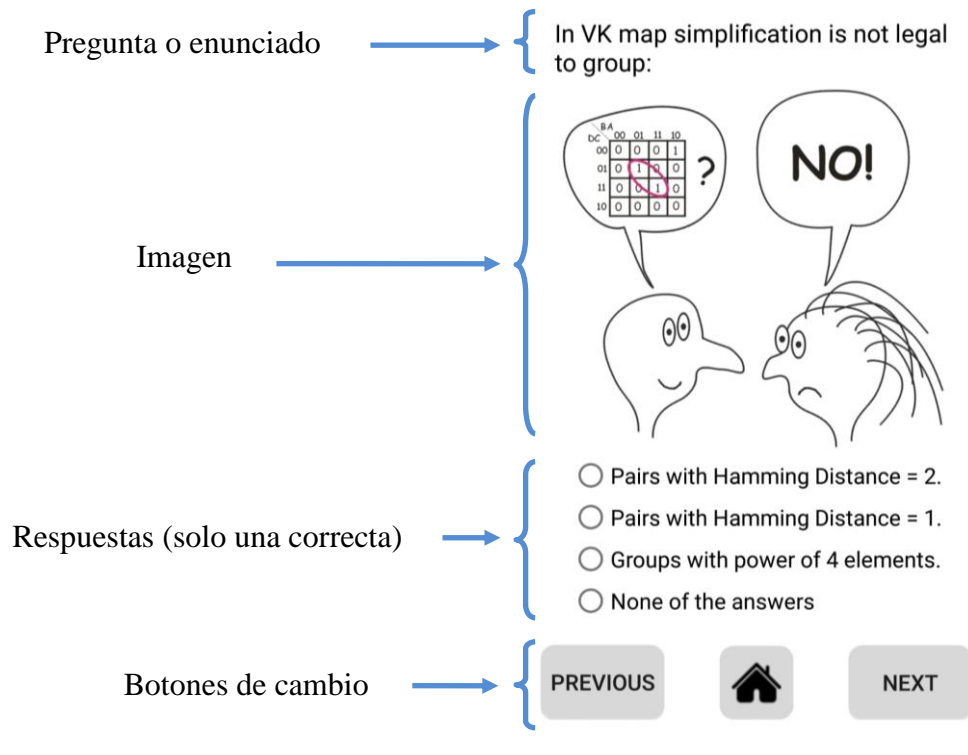


Figura 11. Estructura pregunta test

4.5.1.1 Ampliar imagen

Para poder ampliar las imágenes necesitamos incluir una librería externa.

El sistema de compilación de Gradle en Android Studio facilita la implementación de librerías externas como dependencias. Las dependencias se pueden ubicar de manera automática.

Para agregar una dependencia lo haremos acudiendo al archivo "build.gradle" del proyecto y después, debemos configurar la dependencia como "implementation" en el bloque correspondiente a "dependencies".

En nuestro caso, tenemos la siguiente dependencia binaria remota:

```
implementation 'com.github.chrisbanes:PhotoView:2.1.3'
```

Agregamos la dependencia como hemos mencionado anteriormente en el archivo "build.gradle" que ahora tiene esta forma:

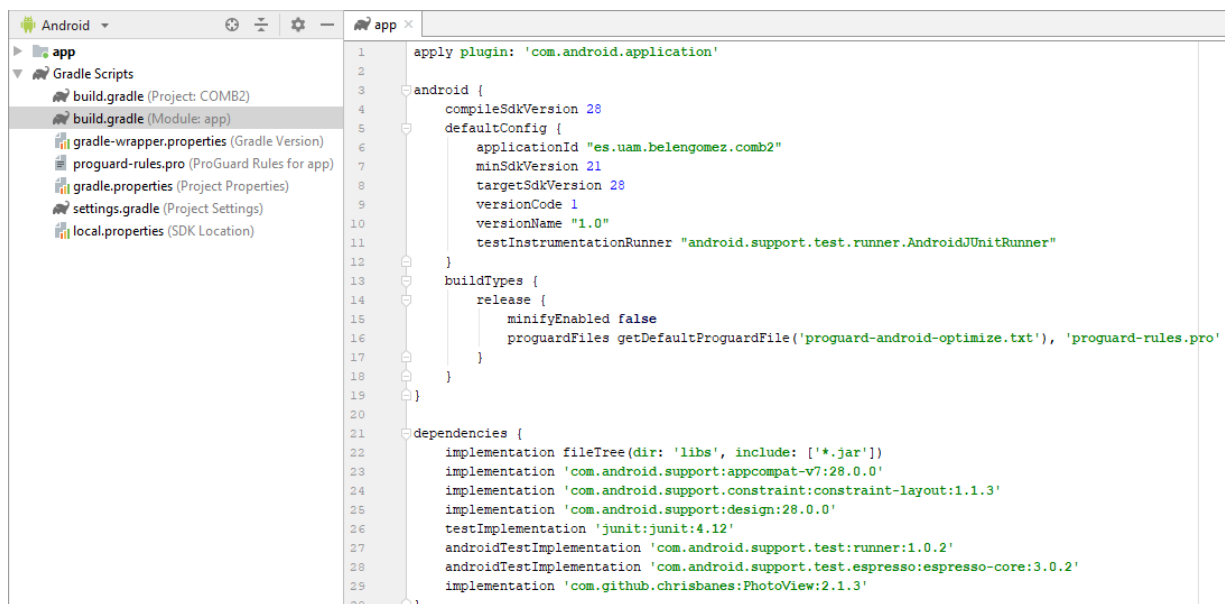


Figura 12. Archivo build.gradle

Una vez está bien implementada la dependencia que implementa la librería para ampliar las imágenes, debemos incluirla en nuestro código java cuando accedamos a las imágenes.

En nuestro caso, las imágenes están almacenadas en un array (all_images), por lo que cada vez que acudimos a buscar la imagen que corresponde a cada pregunta la llamamos como imagen actual (current_image). De esta forma cuando queremos mostrar la imagen de la pregunta, hacemos una llamada a la siguiente función que hace uso de la librería anterior:

```

private void showImage() {
    Drawable mydrawable = getResources().getDrawable(all_images[current_image]);
    photoView.setImageDrawable(mydrawable);
}

```

4.5.1.2 Persistencia de datos

Al tratarse de 15 preguntas de test, nuestra intención es facilitar al usuario que pueda cambiar de respuesta a cualquier pregunta del test que se encuentra realizando, por lo que debemos guardar las respuestas que va marcando para después poder mostrarlas de nuevo cuando el usuario retroceda a preguntas anteriores. Para ello trabajamos con la persistencia de datos, en nuestro caso con Bundle.

Un Bundle guarda información de la actividad en su interior en formato clave-valor. Para trabajar con un Bundle hay que ir a la clase Bundle y si la investigamos en la API, descubrimos que esta clase se hereda de otra (BaseBundle) que le proporciona unos métodos clave para guardar y recuperar la información. Por ejemplo si queremos

guardar información de tipo entero dentro de un Bundle, utilizamos el método `putInt(clave, valor)`, y si queremos recuperar esa información utilizaremos el método `getInt(clave)`.

En nuestro caso usamos Bundle en la función que crea la pregunta. Si es la primera vez que corre la app (`savedInstanceState == null`) la reseteamos en otra función (`startOver()`), pero si no es la primera vez, recuperamos los datos almacenados en Bundle con el método `getInt` como se muestra a continuación.

```
protected void onCreate(Bundle savedInstanceState) {  
  
    super.onCreate(savedInstanceState);  
  
    setContentView(R.layout.activity_primer);  
  
    text_question = findViewById(R.id.text_question);  
    group = findViewById(R.id.answer_group);  
    btn_next = findViewById(R.id.btn_check);  
    btn_prev = findViewById(R.id.btn_prev);  
    photoView = findViewById(R.id.image_view);  
    all_questions = getResources().getStringArray(R.array.all_questions_1);  
  
    if (savedInstanceState == null) {  
        startOver();  
    } else {  
        Bundle state = savedInstanceState;  
        correct_answer = state.getInt(CORRECT_ANSWER);  
        current_question = state.getInt(CURRENT_QUESTION);  
        current_image = state.getInt(CURRENT_IMAGE);  
        answer_is_correct = state.getBooleanArray(ANSWER_IS_CORRECT);  
        answer = state.getIntArray(ANSWER);  
        showQuestion();  
        showImage();  
    }  
}
```

Previamente a recuperar los datos, los almacenamos en la siguiente función (`onSaveInstanceState()`) que utiliza el método `putInt` para guardar los datos.

```
protected void onSaveInstanceState(Bundle outState) {  
  
    super.onSaveInstanceState(outState);  
    outState.putInt(CORRECT_ANSWER, correct_answer);  
    outState.putInt(CURRENT_QUESTION, current_question);  
    outState.putInt(CURRENT_IMAGE, current_image);  
    outState.putBooleanArray(ANSWER_IS_CORRECT, answer_is_correct);  
    outState.putIntArray(ANSWER, answer);  
}
```

Todo este procedimiento se realiza con la intención de que cuando haya cambios en la aplicación, como por ejemplo cambios en la orientación del dispositivo o cuando nuestra aplicación pasa a un segundo plano, se guarden los datos que hemos modificado previamente.

4.5.1.3 Respuesta correcta

El formato que seguimos para introducir las preguntas y las respuestas es el siguiente:

Pregunta; Respuesta 1; Respuesta 2; *Respuesta 3; Respuesta 4

La respuesta correcta irá marcada con * delante (en el ejemplo anterior la respuesta correcta sería la Respuesta 3).

Creamos un array como el siguiente para cada test:

```
<array name="all_questions_1">
  <item>In VK map simplification is not legal to group;:Elements in diagonal.;An amount of
    elements that is power of 2.;Pairs with Hamming Distance = 1.;None of the answers.
  </item>
  <item>In VK map simplification is not legal to group;:Pairs with Hamming Distance = 1.;
    Groups with power of 4 elements.;None of the answers;*Pairs with Hamming Distance = 2.
  </item>
  <item>The result of the grouping is;:DB;*CA;/C./A;/D./B</item>
  <item>In Veitch-Karnaugh maps, the positions are labeled;:00, 01, 10, 11;00, 10, 11, 01;
    *00, 01, 11, 10;None of the answers.
  </item>
  <item>The result of the grouping is;:/D.C;/D.C.A;/D.C./A;*The map is wrong.</item>
  <item>The result of the grouping is;:/A;*/B;1;None of the answers</item>
  <item>The result of the grouping is;:The grouping is illegal.;*/[C+A];D.B;C.A</item>
  <item>The result of the grouping is;:*D./C./A;D./A;D./C.A;D./C./B</item>
  <item>The result of the grouping is;:0;D;*1;D.C.B.A.</item>
  <item>The result of the grouping is;:*None of the answers.;D./C.B./A;/C./A;D.B</item>
  <item>The result of the grouping is;:D./B;*None of the answers.;/A;D./A</item>
  <item>The result of the grouping is;:Optimal.;Illegal.;/C.A+/C.B./A;*Non Optimal.</item>
  <item>The result of the grouping is;:Optimal.;Illegal.;*Non Optimal.;/D.C</item>
  <item>The function F[DCBA] is;:*F = /C.A;F = /A.C;F = C+A;F = /[C+A]</item>
  <item>The function F[DCBA] is;:*F = / [C XOR A];F = C XOR A;F = C + A;F = C . A</item>
</array>
```

Cada ítem del array, es decir cada pregunta completa la debemos descomponer en partes según nos marca los “;” que son los que separan la pregunta y las respuestas.

```
String[] parts = q.split( regex: ";");
```

De esta forma tenemos una cadena con nuestras partes de la pregunta.

Ahora para centrarnos en la búsqueda de la respuesta correcta lo primero que debemos tener en cuenta es que hay que leer a partir del segundo elemento de la cadena que es donde realmente empiezan las respuestas. Después comprobamos si el primer carácter es “*” y si es así ya tenemos localizada la respuesta correcta. Lo último que hay que hacer es comenzar a leer la respuesta correcta a partir del segundo carácter para que no aparezca marcada por pantalla y así no mostrarlo al usuario.

```
String ans = parts[i+1];
if(ans.charAt(0) == '*'){
    correct_answer = j-1;
    ans = ans.substring(1);
}
```

4.5.1.4 Calcular nota

La ponderación de las preguntas en un test es una de las inquietudes más relevantes a la hora de realizar un test, por tanto hemos investigado en diferentes artículos cuál es la manera adecuada de puntuar un test.

Asumimos que las preguntas que no han sido contestadas ni suman ni restan, y las preguntas falladas conllevan una penalización en la nota. La razón de esto es penalizar a aquellos que responden al azar. Esta penalización debe estar relacionada con el número de respuestas posibles.

Esperamos que aquel que responde todo al azar saque un 0 o una puntuación baja. La nota de cada pregunta es una variable aleatoria que puede tener dos valores (obteniendo un 1 si la respuesta es correcta y un valor a determinar si es incorrecta). La nota final del test es la suma de las notas de cada pregunta por lo que también es una variable aleatoria y la nota esperada en el test la obtenemos mediante la esperanza de la nota final.

Por ello, si tenemos N preguntas y obtenemos una nota n en la pregunta i , esta nota puede tomar dos valores (dependiendo de si la respuesta es correcta o no), si acertamos $i=1$ y si fallamos $i=-p$, donde p es lo que resta una respuesta incorrecta.

Cada pregunta tiene k posibles respuestas, y la probabilidad de acertar al responder aleatoriamente es $1/k$, de la misma manera la probabilidad de fallar es $1 - 1/k$. La nota n que obtenemos en la pregunta i viene dada por:

$$n_i = \begin{cases} 1, & \text{con probabilidad } \frac{1}{k} \\ -p, & \text{con probabilidad } 1 - \frac{1}{k} \end{cases}$$

La esperanza mencionada anteriormente es la nota esperada en cada pregunta y se calcula multiplicando valor por probabilidad y sumando los resultados.

$$\begin{aligned} E[n_i] &= 1 \cdot \frac{1}{k} + (-p) \cdot \left(1 - \frac{1}{k}\right) = \frac{1}{k} - p \cdot \frac{k-1}{k} \Rightarrow \\ &\Rightarrow E[n_i] = \frac{1 - p(k-1)}{k} \end{aligned}$$

La nota final del test (NOTA), se define como la suma de las notas de cada pregunta. Como queremos calcular la nota esperada en el test, calculamos la esperanza de NOTA usando la linealidad de la esperanza que determina que la esperanza de la suma es igual a la suma de las esperanzas. Tenemos en cuenta que todas las variables asociadas a la nota de cada pregunta tienen la misma esperanza y obtenemos:

$$E[NOTA] = E \left[\sum_{i=1}^N n_i \right] = \sum_{i=1}^N E[n_i] =$$

$$= \sum_{i=1}^N \frac{1 - p(k-1)}{k} = N \cdot \frac{1 - p(k-1)}{k}$$

Esta expresión se ha calculado partiendo de que respondemos a todas las preguntas de manera aleatoria, ahora bien, si respondiendo de esta manera deberíamos obtener un 0 de nota, igualamos la expresión y despejamos el valor de p:

$$E[NOTA] = 0 \Rightarrow N \cdot \frac{1 - p(k-1)}{k} = 0 \Rightarrow$$

$$\Rightarrow 1 - p(k-1) = 0 \Rightarrow p = \frac{1}{k-1}$$

Finalmente obtenemos que la fórmula adecuada para puntuar el test es:

$$Nota = Aciertos - \frac{Errores}{k-1}$$

En nuestro caso, queremos que nuestra nota este contenida entre 0 y 10, que es como se puntúa la asignatura. Para ello debemos dividir entre el número de preguntas que tiene el test y multiplicar por 10 el resultado. Implementamos para ello el siguiente código:

```
nota = ((correctas-(incorrectas/(n_answers-1)))/n_questions)*10;
```

La anterior fórmula permite obtener notas negativas, para acotar inferiormente nuestra nota a 0 hacemos:

```
if(nota<0)nota=0;
```

Al finalizar cada test aparece un cuadro con los resultados obtenidos, lo implementamos con la ayuda del método builder:

```
AlertDialog.Builder builder = new AlertDialog.Builder( context: this);

builder.setTitle("Results");
String message = String.format("Correct: %d\nIncorrect: %d\nNot selected: %d\nNote: %.2f\n",correctas, incorrectas, nocontestadas, nota);
builder.setMessage(message);
```

El cuadro que aparece en pantalla muestra la nota con la siguiente estructura:

Results
Correct: 11
Incorrect: 2
Not selected: 2
Note: 6,89

Figura 13. Tabla resultados test COMB 2

4.5.1.5 Test aleatorio

Nuestro objetivo es que el alumno razone la respuesta correcta y que no memorice las respuestas por el simple hecho de contestarlas todas bien. Para ello hemos ideado un sistema que modifica la apariencia de las preguntas para que al repetir un mismo test, no tenga el mismo orden. Nuestras respuestas se descolocan de manera aleatoria y el usuario no podrá aprender visualmente las respuestas correctas.

Para asegurarnos de que se organizan lo más aleatoriamente posible acudimos al siguiente código:

```
j = new Random().nextInt( bound: ids_answers.length-1);
```

Esta variable genera un número aleatorio para mover las respuestas el número de veces que indica la variable.

La variable aleatoria se declara en la función startOver() que es llamada cuando se desea repetir un mismo test.

4.5.2 Información

En este apartado hemos creado una interfaz sencilla con una imagen con la información de desarrollo de la aplicación y un botón de regreso al menú principal.

En la imagen debemos tener en cuenta la ampliación de la misma al igual que en las imágenes que se muestran en el test. Para ello implementamos el siguiente código:

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_info);  
  
    PhotoView photoView = findViewById(R.id.photo_view);  
    photoView.setImageResource(R.drawable.dslab);  
}
```

4.5.3 Correo

Para que los usuarios de la aplicación puedan reportar posibles errores que tenga la aplicación y así podamos mejorarla, hemos creado un apartado para que puedan enviar correos electrónicos a las direcciones de los desarrolladores de la aplicación.

La principal característica del diseño es que el usuario solo debe añadir parte del asunto y el cuerpo del mensaje, con esto conseguimos evitar que el usuario añada mal las direcciones de correo y nos aseguramos de que se van a enviar correctamente. Implementamos en el código las direcciones de los creadores y en el asunto fijamos una cabecera (“App COMB 2”) para facilitar la lectura de los correos y así localizarlos fácilmente en la bandeja de entrada.

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_correo);  
  
    mSubjectEt = findViewById(R.id.subjectEt);  
    mMessageEt = findViewById(R.id.messageEt);  
    mSendEmailBtn = findViewById(R.id.sendEmailBtn);  
  
    mSendEmailBtn.setOnClickListener(new View.OnClickListener() {  
        @Override  
        public void onClick(View v) {  
            String subject = mSubjectEt.getText().toString().trim();  
            String message = mMessageEt.getText().toString().trim();  
            String subjectComplete = "App COMB 2: " + subject;  
            String[] to = { "belen.gomezh@estudiante.uam.es" };  
            String[] cc = { "eduardo.boemo@uam.es" };  
            sendEmail(to, cc, subjectComplete, message);  
        }  
    });  
}
```

El sistema usará los correos que disponga el dispositivo, ya que el correo se debe enviar desde un correo que disponga el usuario previamente, además esto acelera el proceso ya que probablemente el usuario tenga iniciada la sesión de su correo en su dispositivo.

Por último desde otra función accedemos al correo del usuario y todos los contenidos que se han añadido a nuestra interfaz se posicionan en los lugares correspondientes a la interfaz del correo que se utilice.

4.6 Adaptación pantallas

4.6.1 ScrollView

La limitación de la pantalla frente a los objetos que queremos mostrar en ella hace que tengamos que hacer uso de herramientas como ScrollView. Lo que conocemos como “hacer scroll” no es más que una forma de poder desplazar objetos en la pantalla para hacerlos visibles.

Para poder desplazar todo los objetos, debemos hacer un diseño en árbol jerárquico con ellos, de tal forma que ScrollView contenga todos los elementos que debe mostrar la pantalla. En nuestro caso tenemos problemas con el scroll vertical ya que no se pueden adaptar los elementos a las dimensiones de la altura que tome el dispositivo, los elementos tienen unas dimensiones que se deben respetar para mantener la coherencia del diseño.

En nuestro diseño de los test planteamos el siguiente esquema:

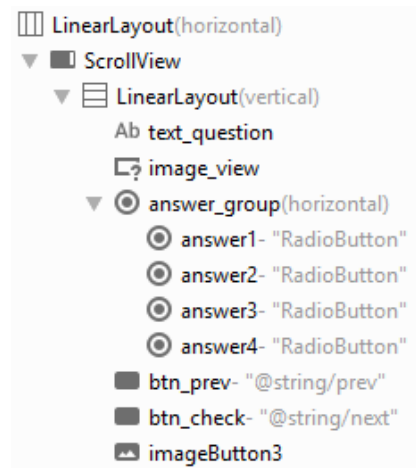


Figura 14. Estructura diseño pregunta test

La vista que tiene nuestro diseño, adapta de la siguiente manera los elementos:

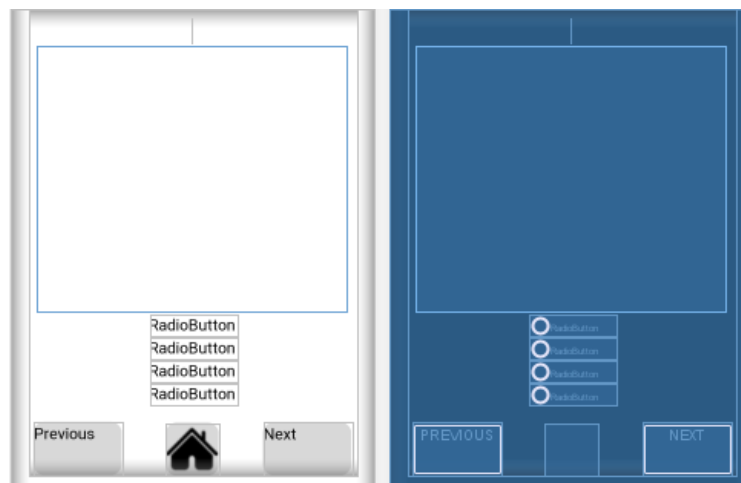


Figura 15. Vista diseño pregunta test

4.6.2 Wrap_content y Match_parent

Para asegurarnos de que nuestro diseño sea flexible y se adapte a los diferentes tamaños de las pantallas de los dispositivos utilizamos wrap_content y match_parent para el ancho y alto de algunos componentes.

- Wrap_content: con él se fija el ancho o el alto de la vista en el tamaño mínimo necesario para adaptar el contenido dentro de esa vista.
- Match_parent: con él hacemos que el componente se extienda hasta llegar a adaptarse al tamaño de la primera vista.

Al usar estos dos valores en vez de establecer tamaños fijos, hacemos que las vistas de nuestro proyecto tome el espacio requerido en cada instante, así cuando se producen rotaciones de pantalla en las que el dispositivo toma medidas de pantalla de alto y ancho muy diferentes según el sentido, logramos una vista uniforme y razonable.

En nuestro caso el tamaño de los botones los fijamos para siempre tener una interfaz igualada, sin embargo con los textos del enunciado y los botones, adaptamos a la vista usando wrap_content, en el caso del ScrollView y el LinearLayout que contienen todos los elementos, se diseñan con match_parent para que contengan toda la pantalla del dispositivo. Con esto conseguimos hacer scroll en toda la pantalla y que los textos se adecuen al ancho de la pantalla haciéndolos más legibles.

5 Integración, pruebas y resultados

5.1 Publicar App

En el momento en que nuestra aplicación es válida para instalar en dispositivos y se ha comprobado que todo funciona correctamente, procedemos a publicar nuestra App en la tienda de aplicaciones móviles Google Play Store.

5.1.1 Generar fichero APK

El primer paso que debemos realizar a la hora de publicar nuestra aplicación es preparar el fichero APK (Android Application Package), este fichero contiene toda la información de nuestra app.

Para generar el fichero APK, seguimos los siguientes pasos:

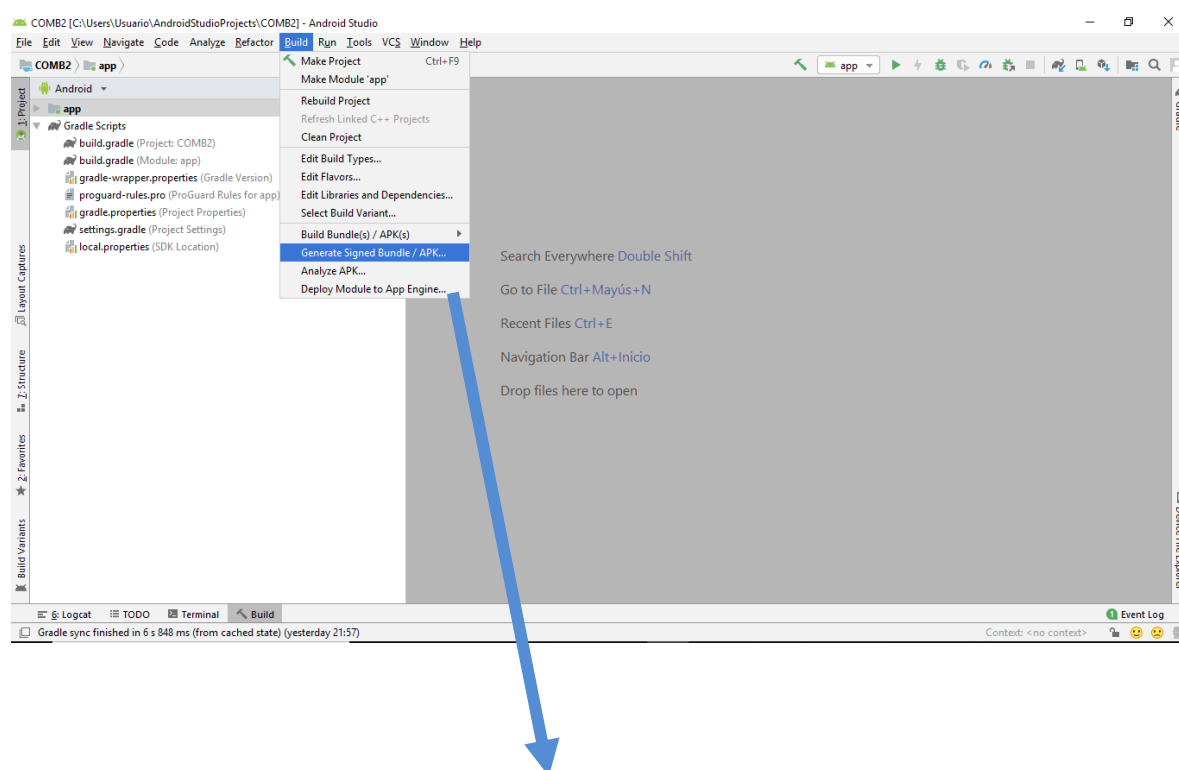


Figura 16. Primer paso generar fichero APK

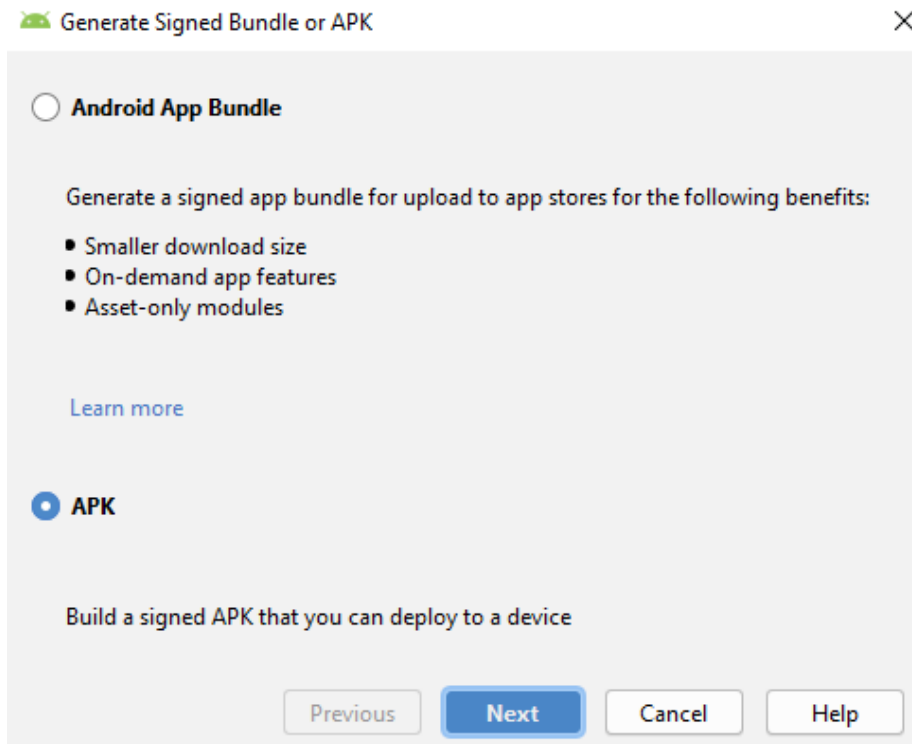


Figura 17. Segundo paso generar fichero APK

Se elige la opción de release porque queremos lanzar la aplicación, si quisiéramos probarla solamente elegiríamos la opción de debug.

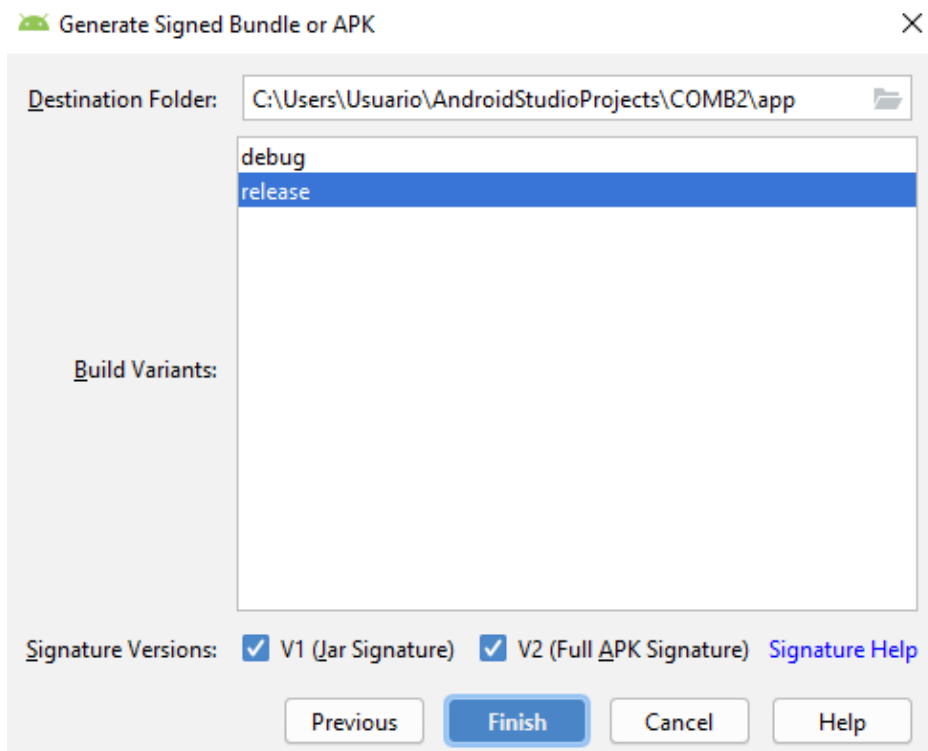


Figura 18. Tercer paso generar fichero APK

Para publicar la aplicación es necesario firmarla con un certificado digital, este proceso es muy importante ya que la firma es lo que nos va a permitir realizar modificaciones posteriormente de manera segura.

Para firmarla debemos crear un Key store. Una vez creado, tan solo necesitaremos proporcionar la ubicación del archivo, la contraseña y el alias.

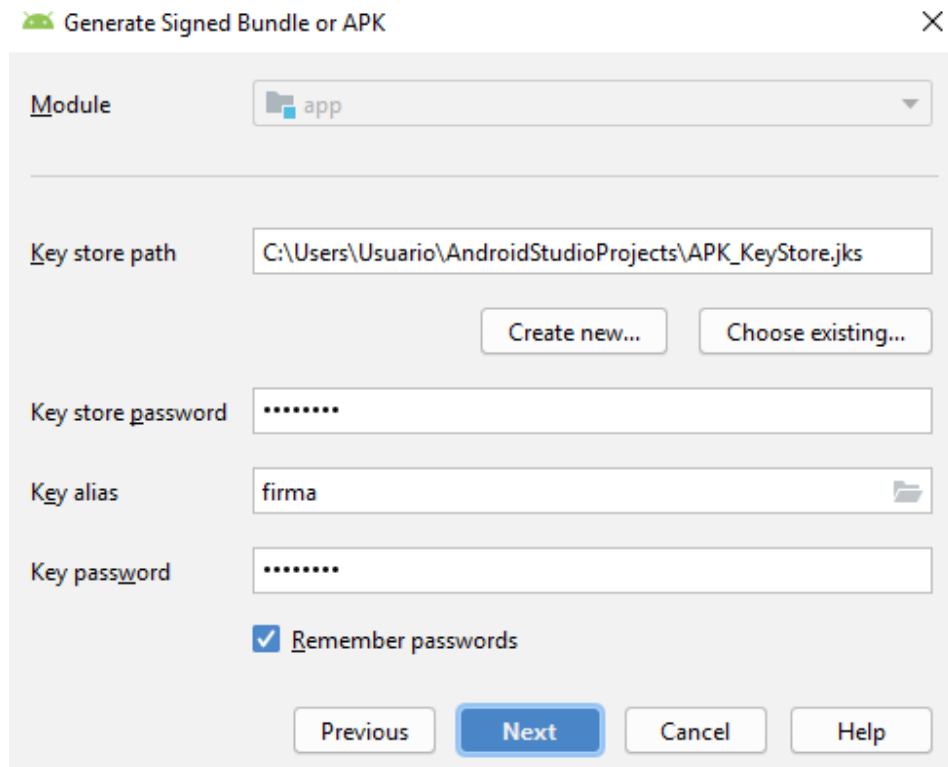


Figura 19. Cuarto paso generar fichero APK (firma)

Después de firmar nuestro archivo, se generará nuestro archivo APK deseado. Este fichero se encuentra dentro de la carpeta app\release de nuestro proyecto.

5.1.2 Cuenta desarrollador en Google Play

Para poder publicar una aplicación en Google Play debemos tener una cuenta en Google, además es necesaria una cuenta de email para poder enviar la aplicación a Google Play y poder así acceder a la Consola de Android Developers.

Una vez podemos acceder a la Consola de Android Developers, podemos registrarnos como desarrolladores de Google y pagando una sola cuota de 25\$ podremos publicar todas las aplicaciones que deseemos sin tener que hacer más pagos, a diferencia de Apple Store donde la cuota es anual y de un valor de 99\$.

En nuestro caso haremos uso de la cuenta del laboratorio de la universidad ("DSLAb UAM").

5.1.3 Publicación en Google Play

Para agregar una aplicación en nuestra Consola de Android, añadimos una aplicación, elaboramos la ficha con todos los datos que se nos solicita y agregamos el material multimedia que necesita Google para mostrar nuestra aplicación en su tienda virtual.

Si no queremos hacer de una sola vez este proceso, podemos crear la plantilla y dejarla en estado de “no publicada” para poder seguir trabajando en ella en otro momento. O también se pueden subir actualizaciones de la aplicación con nuevas versiones del fichero APK, sin tener que repetir todo el proceso de creación de la ficha en Play Store.

Debemos cumplir los requisitos que nos plantean para crear la ficha de nuestra app, como el número máximo de contenido en los textos, los tamaños exactos en las imágenes y demás contenido.

Para nuestra aplicación, al tener los contenidos en inglés, se desarrolla la ficha de Play Store en este idioma. Como buscamos que el mayor número de personas puedan beneficiarse de este proyecto se configura para que sea una app gratuita y disponible hasta en 149 países.

Para que nuestra aplicación se pueda ofrecer al público, debemos cumplir una serie de requisitos en el contenido (no contener contenido sexual, violento, etc.). Para ello es necesario responder una serie de preguntas y así obtener las etiquetas de clasificación de contenido adecuadas. En nuestro caso al tratarse de una aplicación educativa es apta para todos los públicos. Obtenemos los siguientes distintivos de clasificación de contenido:



Figura 20. Clasificación aplicada a COMB 2

5.2 *Análisis de la aplicación*

Google nos permite tener acceso a las estadísticas de descargas de nuestra aplicación, esto es de gran ayuda para saber cómo recibe el público nuestra app.

Como nuestro objetivo es reforzar el estudio de los estudiantes de Circuitos Electrónicos Digitales, analizaremos estas estadísticas a lo largo del siguiente curso cuando los alumnos se beneficiarán de esta aplicación.

6 Conclusiones y trabajo futuro

6.1 Conclusiones

Tras dar por finalizado el proyecto, se ha llegado a la conclusión de que se han cumplido los objetivos marcados al principio de este trabajo, y se han implementado mejoras adicionales que no estaban planeadas al inicio.

El objetivo inicial del proyecto era aprovechar el auge de la tecnología en las aulas para crear una herramienta útil que pudiera servir de ayuda para los alumnos de la asignatura de Circuitos Electrónicos Digitales. Esta aplicación ofrece el material educativo necesario para cubrir parte del temario que trata.

Se ha desarrollado un código reducido, óptimo y eficaz que cumple los requisitos marcados, realiza 4 test de igual complejidad a la que los alumnos se enfrentarán en las pruebas que se realizarán en el aula, siguiendo el mismo criterio de corrección que se lleva a cabo en la asignatura.

Desde el punto de vista formativo, el desarrollo de esta app ha permitido el aprendizaje de contenidos – no vistos durante la carrera - tales como:

- Desarrollo en SO Android.
- Uso del programa Android Studio.
- Lenguaje de programación Java.
- Lenguaje de programación XML.
- Mecanismo de publicación de aplicaciones en la plataforma Google Play Store.

6.2 Trabajo futuro

La aplicación es susceptible a diversas modificaciones. Se ha implementado un código simplificado de forma que se pueden modificar todas las preguntas de una manera sencilla sin tener que modificar la estructura de la aplicación.

En cuanto a temas de diseño se podrían añadir mejoras como por ejemplo contenido audiovisual diferente (videos, audios, etc.) hasta donde esté contemplado en el temario de la asignatura. También podríamos añadir una opción para que los propios alumnos pudieran enviar sus propuestas de preguntas, de esta forma se fomentaría la creatividad de los alumnos en temas de electrónica digital.

Respecto al contenido de la aplicación, se podría incorporar diversas herramientas que no se han planteado en este trabajo por formar parte de proyectos ya realizados pero con los que nuestra aplicación sería mucho más completa, como incluir teoría, calculadoras, conversores, etc.

En el tema comercial, la creación de una aplicación Android nos ofrece diversas maneras de rentabilizar una app, mediante la plataforma de Play Store se nos ofrece la posibilidad de obtener beneficios económicos con la incorporación de publicidad. Además podemos exponer nuestra aplicación en otras plataformas como las redes sociales.

Si queremos acceder a más usuarios, aparte de ponerla en disponibilidad de 149 países, se puede traducir a más idiomas.

Referencias

Las principales referencias utilizadas en este trabajo se encuentran *on-line*. Una lista de ellas es:

[1] <https://play.google.com/store/apps>

[2] <https://www.youtube.com/watch?v=8wpmRNBLWRI>

[3] <https://www.youtube.com/watch?v=GM8sc0ZjopU>

[4] <http://codictados.com>

[5] <https://www.youtube.com/watch?v=q-N0unUiLKY&list=PLU8oAlHdN5Bkn-KS1sRFISEnXXcAtAJ9P&index=57>

[6] <https://www.youtube.com/watch?v=5nn3Zyyf6nk&index=23&list=PLU8oAlHdN5Bkn-KS1sRFISEnXXcAtAJ9P>

[7] <http://www.hermosaprogramacion.com>

[8] <https://es.statista.com>

[9] <https://codelabs.developers.google.com>

[10] <http://stackoverflow.com>

[11] <https://developer.android.com>

[12] Miguel Ángel Morales. “Las matemáticas de la fórmula de puntuación de exámenes test”, El País, Febrero 2017.

[13] J.L González-Santander y G. Martín. “Análisis para la calificación de pruebas tipo test multi-respuesta”, Nereis, Noviembre 2010.

Glosario

DSLAb	Digital System Laboratory
EPS	Escuela Politécnica Superior
UAM	Universidad Autónoma de Madrid
SO	Sistema Operativo
API	Application Programming Interface
APK	Android Application Package
SDK	Software Development Kit
XML	eXtensible Markup Language

Anexos

A Manual de instalación

Para poder instalar la aplicación en un dispositivo debemos acudir a la tienda de aplicaciones de Google Play Store.

Hemos creado un código QR con el que los usuarios pueden acceder fácilmente a la página de instalación de la aplicación, tan solo es necesario disponer de cualquier lector de códigos QR.

La dirección web a la que accede el código QR y a la que debemos acceder para descargarla es:


<https://play.google.com/store/apps/details?id=es.uam.belengomez.comb2>

- Código QR de COMB 2:



Figura 21. Código QR COMB 2

- Play Store COMB 2:



COMB 2

DSLab UAM Educación

PEGI 3

Añadir a la lista de deseos

Instalar

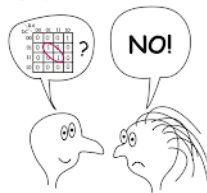
VEITCH-KARNAUGH MAPS

COMBINATIONAL CIRCUITS I

COMBINATIONAL CIRCUITS II

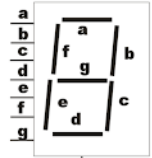
ELECTRICAL CONSIDERATIONS

In VK map simplification is not legal to group.



☐ None of the answers
☐ Pairs with Hamming Distance = 2.
☐ Pairs with Hamming Distance = 1.
☐ Groups with power of 4 elements.

Writing number 3 in this display requires:



Common Anode
☐ abcdefg = 0000110
☐ abcdefg = 1001111
☐ abcdefg = 1111001
☐ abcdefg = 0110000

The output:

☐ /C/B
☐ /C X
☐ C XO
☐ /C/B

PREVIOUS

NEXT

PREVIOUS

NEXT

PREVIOUS

¿Quieres traducir la descripción al Español (España) con el Traductor de Google?

Traducir

This app implements a test of 60 questions to prepare the exam of Combinational Blocks (Unit 1) of Digital Electronic Circuits (CED) of the School of Engineering of the Universidad Autónoma de Madrid.

The app is divided in four sections: a) Veitch-Karnaugh graphical simplification; b) Basic combinational blocks; c) MUX, DECOD, and other circuits; and d) Electrical details of digital circuits.

INFORMACIÓN ADICIONAL

Actualizada 16 de mayo de 2019	Tamaño 4,5M	Descargas 10+
Versión actual 1.0	Requiere Android 5.0 y versiones posteriores	Clasificación de contenido PEGI 3 Más información
Permisos Ver detalles	Informe Marcar como inapropiado	Ofrecida por Google Commerce Ltd
Desarrollador Visitar sitio web dslab.uam@gmail.com		

Figura 22. Play Store COMB 2

B Manual de usuario

Cuando el usuario inicie la aplicación por primera vez se encontrará con lo siguiente:

Splash informativo de la aplicación con una duración de 7 segundos. Después aparecerá el menú de la aplicación, donde nos encontramos con 4 test de 15 preguntas y dos botones adicionales que nos llevan a la información de la aplicación y a un apartado para enviar correos a los desarrolladores. Estos bloques están más detallados en el apartado 3 de esta memoria (Diseño/Bloques aplicación).

Se ha desarrollado una interfaz muy intuitiva y flexible por lo que es accesible para todos los usuarios, aunque carezcan de conocimientos informáticos o similares.

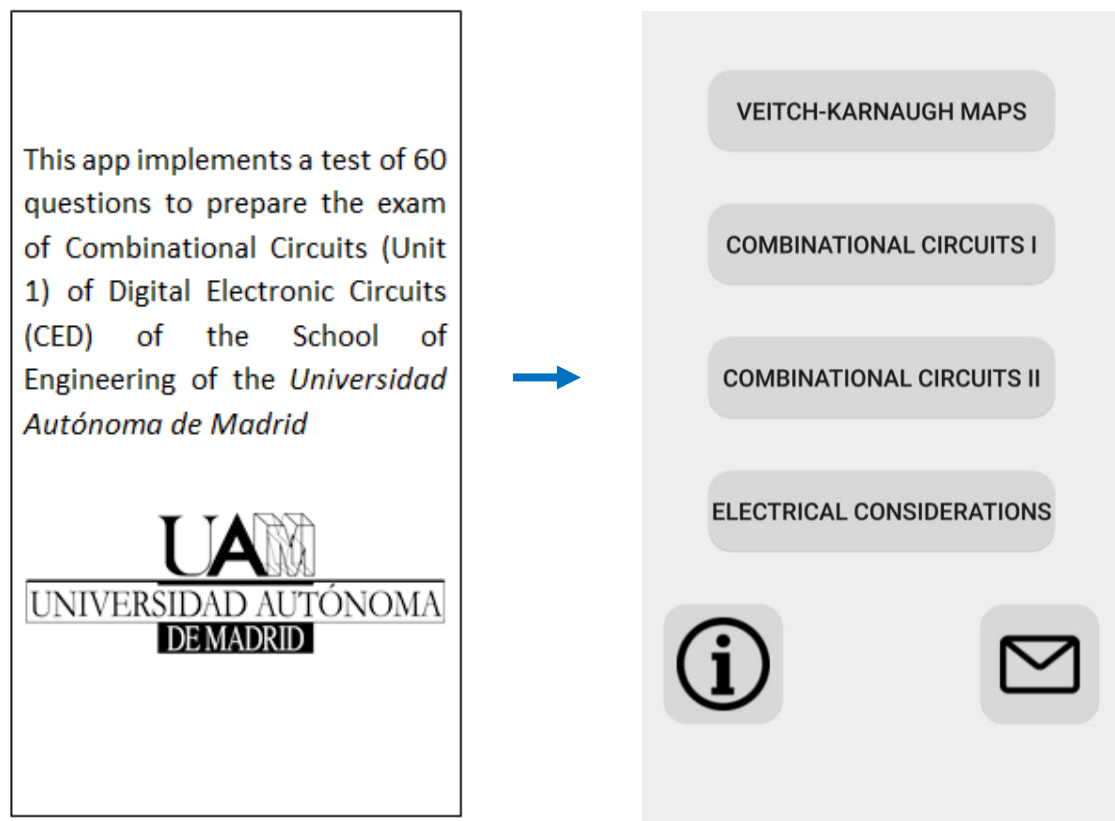
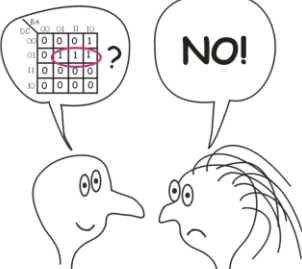
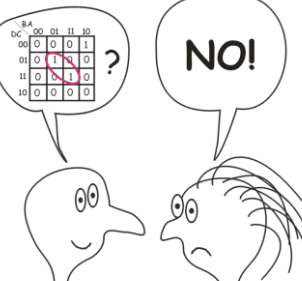
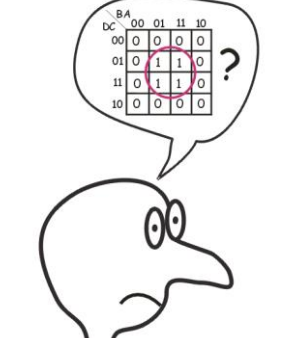
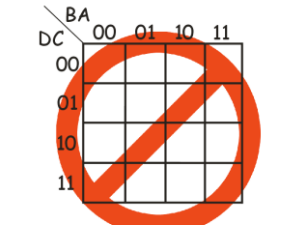
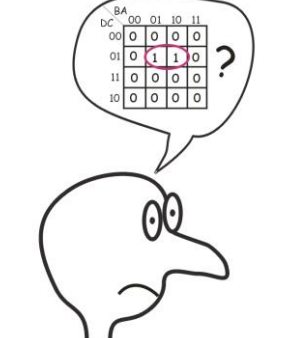
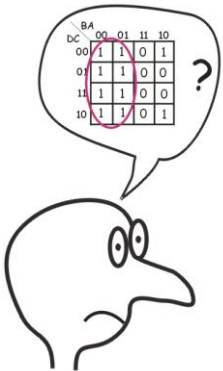
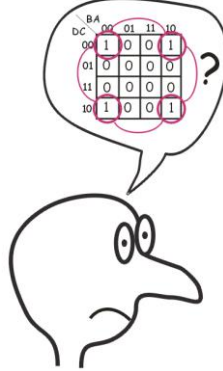
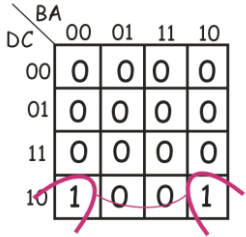
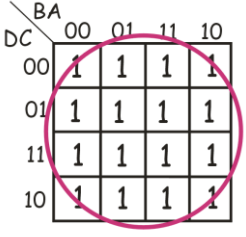
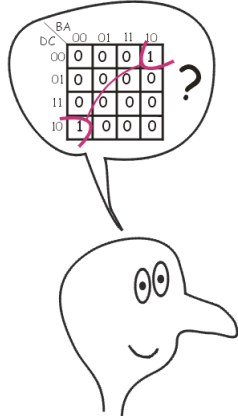
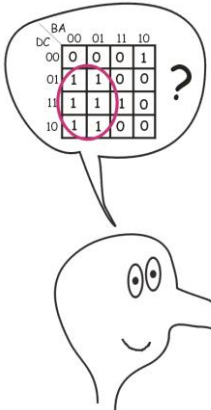
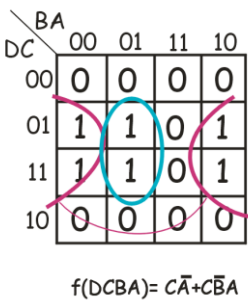
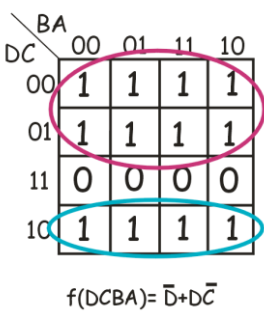
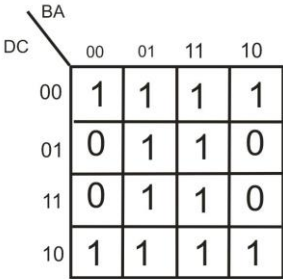
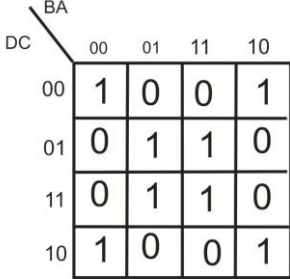



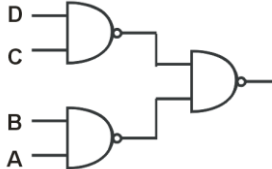
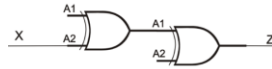
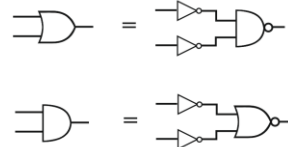
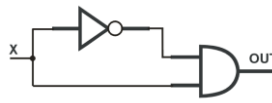
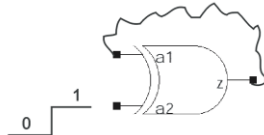
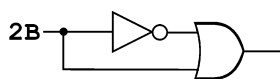
Figura 23. Inicio COMB 2

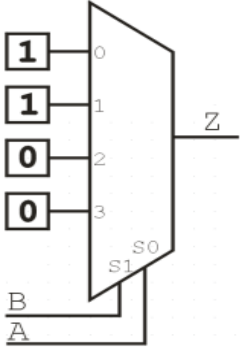
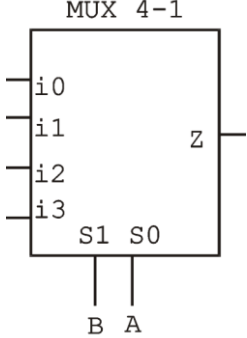
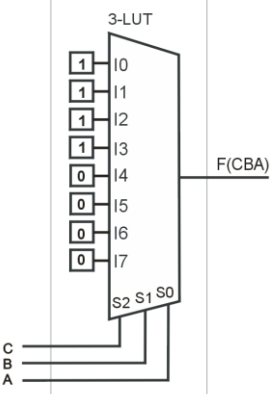
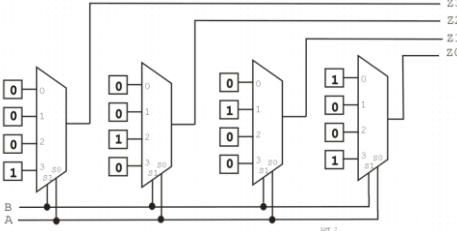
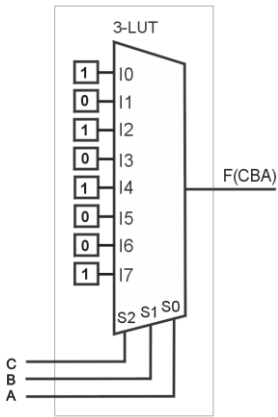
C Preguntas test

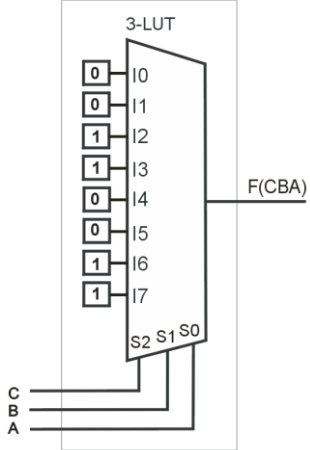
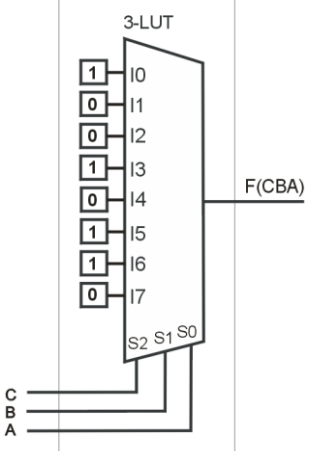
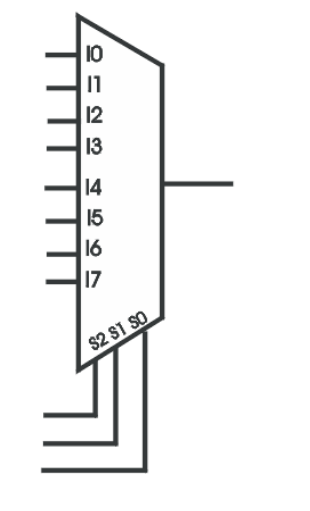
Mapas Veitch-Karnaugh		
Enunciado	Imagen	Respuestas
In VK map simplification is not legal to group:		<ul style="list-style-type: none"> • Elements in diagonal. • An amount of elements that is power of 2. • Pairs with Hamming Distance = 1. • None of the answers.
In VK map simplification is not legal to group:		<ul style="list-style-type: none"> • Pairs with Hamming Distance = 2. • Pairs with Hamming Distance = 1. • Groups with power of 4 elements. • None of the answers
The result of the grouping is:		<ul style="list-style-type: none"> • CA • DB • /C./A • /D./B
In Veitch-Karnaugh maps, the positions are labeled:		<ul style="list-style-type: none"> • 00, 01, 11, 10 • 00, 01, 10, 11 • 00, 10, 11, 01 • None of the answers.
The result of the grouping is:		<ul style="list-style-type: none"> • The map is wrong. • /D.C • /D.C.A • /D.C./A

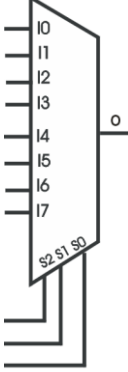
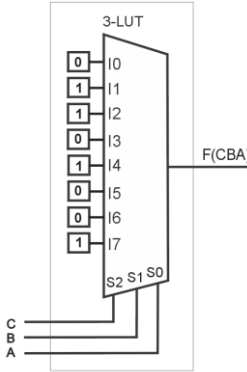
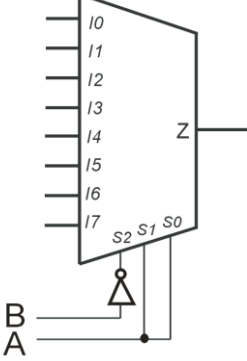
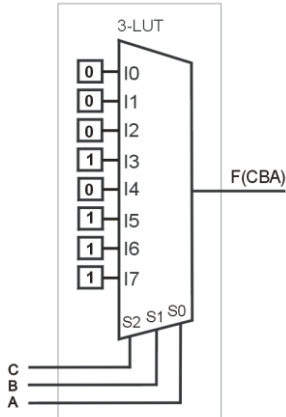
<p>The result of the grouping is:</p>		<ul style="list-style-type: none"> • /B • /A • 1 • None of the answers
<p>The result of the grouping is:</p>		<ul style="list-style-type: none"> • /[C+A] • The grouping is illegal. • D.B • C.A
<p>The result of the grouping is:</p>		<ul style="list-style-type: none"> • D./C./A • D./A • D./C.A • D./C./B
<p>The result of the grouping is:</p>		<ul style="list-style-type: none"> • 1 • 0 • D • D.C.B.A.
<p>The result of the grouping is:</p>		<ul style="list-style-type: none"> • None of the answers. • D./C.B./A • /C./A • D.B

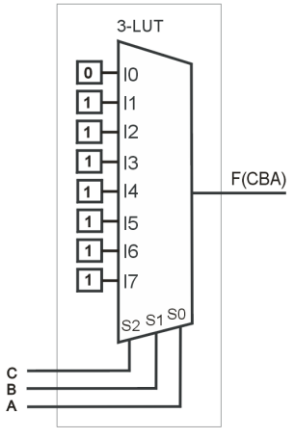
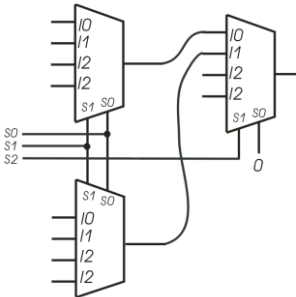
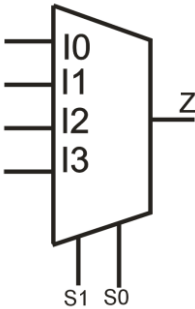
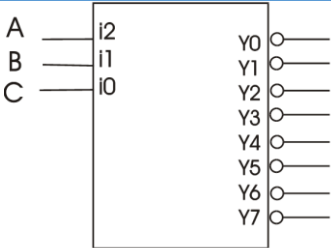
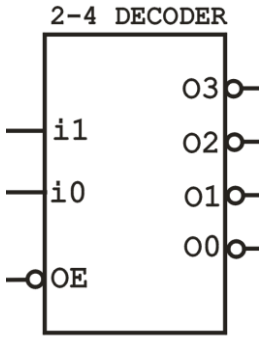
<p>The result of the grouping is:</p>		<ul style="list-style-type: none"> • None of the answers. • D./B • /A • D./A
<p>The result of the grouping is:</p>	 $f(DCBA) = C\bar{A} + C\bar{B}A$	<ul style="list-style-type: none"> • Non Optimal. • Optimal. • Illegal. • /C.A+/C.B./A
<p>The result of the grouping is:</p>	 $f(DCBA) = \bar{D} + D\bar{C}$	<ul style="list-style-type: none"> • Non Optimal. • Optimal. • Illegal. • /D.C
<p>The function F[DCBA] is:</p>		<ul style="list-style-type: none"> • $F = /C . A$ • $F = /A . C$ • $F = C + A$ • $F = / [C+A]$
<p>The function F[DCBA] is:</p>		<ul style="list-style-type: none"> • $F = / [C \text{ XOR } A]$ • $F = C \text{ XOR } A$ • $F = C + A$ • $F = C . A$

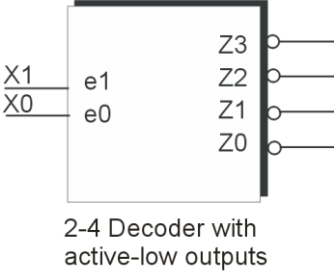
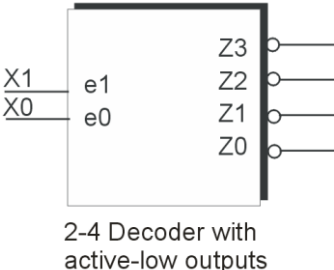
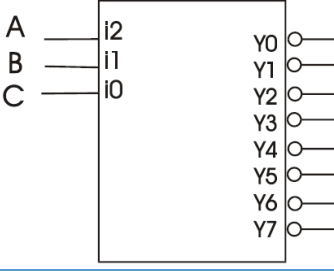
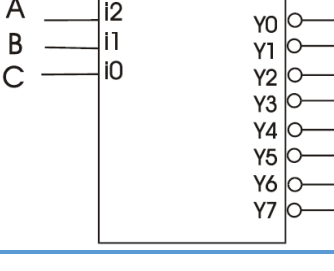
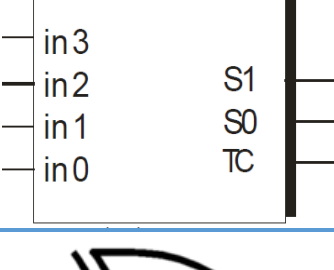

Circuitos Combinacionales I		
Enunciado	Imagen	Respuestas
The following gate is a:		<ul style="list-style-type: none"> • 1x1 bit binary multiplier. • 2-input OR. • 2-input NAND. • None of the answers.
The output of the circuit is:		<ul style="list-style-type: none"> • $D.C+B.A$ • $[D+C].[B+A]$ • $/[D.C]+/[B.A]$ • $/[[D+C].[B+A]]$
The right answer is:		<ul style="list-style-type: none"> • Z=1 if X=1, A1=1, and A2=1. • Z=0 if X=1, A1=1, and A2=1. • Z=1 if X=0, A1=0, and A2=0. • Z=0 if X=0, A1=1, and A2=0.
In the following figure:		<ul style="list-style-type: none"> • Above and below equivalences are true. • Above equivalence is true, below false. • Above equivalence false, below true. • Above and below equivalences are false.
The output of the AND:		<ul style="list-style-type: none"> • Can reach 1 if signal x pass from 0 to 1. • Can reach 1 if signal x pass from 1 to 0. • Is always 0. • Is always 1.
Z and a1 input are accidentally connected. If a2 goes to 1, then:		<ul style="list-style-type: none"> • Z change between 0 and 1 indefinitely. • Z = 0 • Z = 1 • None of the answers.
The output of the circuit is the most famous phrase of:		<ul style="list-style-type: none"> • W. Shakespeare. • G. Boole. • A. De Morgan. • R. Hamming.


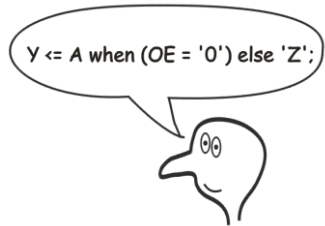
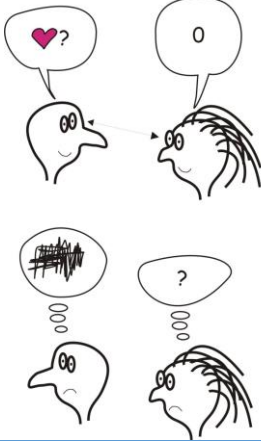
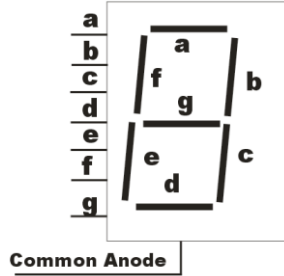
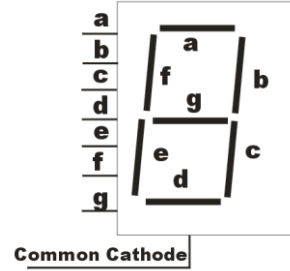
<p>The 4-1 MUX maps:</p>		<ul style="list-style-type: none"> • $Z = \neg A$ • $Z = \neg[A \cdot B]$ • $Z = \neg B$ • $Z = \neg[B + A]$
<p>In the 4-1 multiplexer:</p>		<ul style="list-style-type: none"> • $Z = 1$ if $BA=10$ and $i2=1$ • $Z = 0$ if $BA=10$ and $i2=1$ • $Z = 1$ if $AB=10$ and $i2=1$ • $Z = 0$ if $AB=10$ and $i2=0$
<p>The output of the 8-1 MUX is:</p>		<ul style="list-style-type: none"> • $\neg C$ • $\neg B$ • $\neg A$ • $C + B + A$
<p>The four 4-1 MUX based circuit is a:</p>		<ul style="list-style-type: none"> • 2-bit square elevator. • 2x2 bit multiplier. • 2-bit adder. • None of the above.
<p>In the 8-1 MUX, F(CBA) is:</p>		<ul style="list-style-type: none"> • $\neg A$ • $\neg B$ • $\neg C$ • $C \text{ XOR } B \text{ XOR } A$


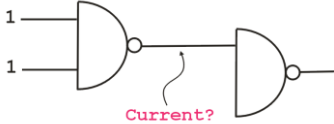
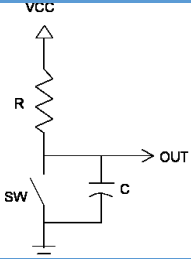
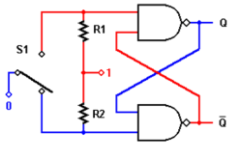
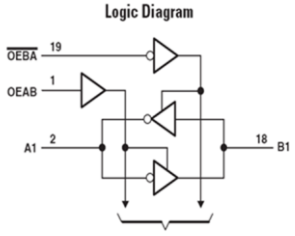
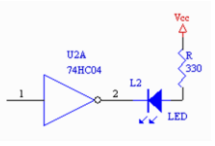
<p>In the 8-1 MUX, F[CBA] is</p>		<ul style="list-style-type: none"> • $\neg B$ • $\neg C$ • $\neg A$ • CB
<p>In the 8-1 MUX, F[CBA] is</p>		<ul style="list-style-type: none"> • $\neg[C \text{ XOR } B \text{ XOR } A]$ • $[C \text{ XOR } B \text{ XOR } A]$ • $\neg C/BA + \neg CB/A + C/B/A + CBA$ • $\neg C/B/A + CBA + \neg CB/A + \neg CBA$
<p>A 16-1 MUX can be efficiently constructed using:</p>		<ul style="list-style-type: none"> • Only three 8-1 MUXs. • Only four 8-1 MUXs. • Only three 8-1 MUXs plus an inverter. • Only four 8-1 MUXs plus an inverter.

Circuitos Combinacionales II		
Enunciado	Imagen	Respuestas
In order to implement a 4-1 using a 8-1 MUX:		<ul style="list-style-type: none"> S2 must be grounded and I[7:4] must be left unconnected. S2 and I[3:0] must be grounded. S2 must be connected to logic 1 and I[7:4] must be grounded. S2 and I[7:4] must be left unconnected.
The output of the 8-1 MUX is:		<ul style="list-style-type: none"> $C \text{ XOR } B \text{ XOR } A$ $\neg[C \text{ XOR } B \text{ XOR } A]$ $\neg C/BA + CB/A + CB/A + CBA$ $\neg C/B/A + C/BA + C/B/A + CBA$
$Z = A \text{ XOR } B$ if:		<ul style="list-style-type: none"> I0=0, I3= 1, I4=0, I7=1, and the other input connected to 0. I0=0, I2= 1, I4=0, I7=1, and the other input connected to 0. I2=0, I3= 1, I4=0, I7=1, and the other input connected to 1. I0=0, I3= 1, I5=0, I7=1, and the other input connected to 1.
The output of the 8-1 MUX is:		<ul style="list-style-type: none"> $CA + CB + BA$ $C \text{ XOR } B \text{ XOR } A$ $[C + A][C + B][A + B]$ C

<p>The output of the 8-1 MUX is:</p>		<ul style="list-style-type: none"> • $C+B+A$ • CBA • $\neg[CBA]$ • $\neg[C+B+A]$
<p>The following arrange of 4-1 MUXs implements:</p>		<ul style="list-style-type: none"> • None of the answers. • 8-1 MUX • 8-1 CODER • 8-1 DEMUX
<p>In the following 4-1 MUX, If Z is utilized as input and I[3:0] as output, it can be obtained a:</p>		<ul style="list-style-type: none"> • None of the answer. • 1-4 demultiplexer. • 1-4 decoder. • 1-4 coder.
<p>In the 3-8 decoder with active-low outputs:</p>		<ul style="list-style-type: none"> • $Y7 = 0$ if $ABC = 111$. • $Y7 = 1$ if $ABC = 111$. • $Y4 = 0$ if $CBA = 100$. • $Y4 = 1$ if $CBA = 100$
<p>In the following 2-4 decoder:</p>		<ul style="list-style-type: none"> • $O2 = \neg i1 + i0 + OE$ • $O2 = i1 + \neg i0 + OE$ • $O2 = \neg[i1 i0] + OE$ • $O2 = [i1 i0] + OE$

In the decoder of the figure:	 <p>2-4 Decoder with active-low outputs</p>	<ul style="list-style-type: none"> $Z0 = X1 + X0$ $Z1 = \neg X1 + X0$ $Z2 = X1 + \neg X0$ $Z3 = \neg X1 + \neg X0$
In the decoder of the figure:	 <p>2-4 Decoder with active-low outputs</p>	<ul style="list-style-type: none"> $Z3 = X1 \text{ NAND } X0$ $Z2 = X1 \text{ NOR } X0$ $Z1 = X1 \text{ OR } X0$ $Z0 = Z1 = X1 \text{ AND } X0$
In the 3-8 decoder of the figure:		<ul style="list-style-type: none"> $Y3=0$ if $ABC = 011$ $Y3=0$ if $ABC = 100$ $Y3=1$ if $CBA = 011$ None of the answers.
In the 3-8 decoder of the figure:		<ul style="list-style-type: none"> $Y4 = \neg I2 + I1 + I0$ $Y4 = \neg I2 \cdot I1 \cdot I0$ $Y4 = I2 + \neg I1 + \neg I0$ $Y4 = I2 \cdot \neg I1 \cdot \neg I0$
In the priority 4-2 coder of the figure (in3 max priority):		<ul style="list-style-type: none"> None of the answers $S1S0 = 10$ if $\text{in}[3:0] = 1100$ $S1S0 = 10$ if $\text{in}[3:0] = 0011$ $S1S0 = 01$ if $\text{in}[3:0] = 1011$
The output of the XOR of n bits is 1 if:		<ul style="list-style-type: none"> The number of bits equal to 1 is even. The number of bits equal to 1 is odd. The number of bits is even. The number of bits is odd.

Electrónica básica		
Enunciado	Imagen	Respuestas
The “ALS” in the seriegraphy means:		<ul style="list-style-type: none"> • Advanced Low-power Schottky. • Advanced logic Set. • Antistatic logic Setup. • A-serie Logic Set
3-state output is:		<ul style="list-style-type: none"> • The output can be programmed as a high-impedance. • The output logic level corresponding to "1" is 3 volts. • The slew-rate of the output can be selected. • Output level shifter.
Active-Low is:		<ul style="list-style-type: none"> • A signal that is activated with a logical 0. • A signal that is activated with exactly 0 Volt. • A signal that is slow. • High activity signal.
Writing number 3 in this display requires:		<ul style="list-style-type: none"> • abcdefg = 0000110 • abcdefg = 1111001 • abcdefg = 0110000 • abcdefg = 1001111
Writing number 3 in this display requires:		<ul style="list-style-type: none"> • abcdefg = 1111001 • abcdefg = 0000110 • abcdefg = 0110000 • abcdefg = 1001111

Regarding standard chips:		<ul style="list-style-type: none"> Unused outputs should be left unconnected. Unused inputs should be left unconnected. Unused outputs should be always grounded. Unused inputs should be always grounded.
If a connected output pin goes to 0		<ul style="list-style-type: none"> the current enters the pin. the current outputs the pin. the current is 0 mA. There is a short-circuit
The circuit is useful for an:		<ul style="list-style-type: none"> Active-low reset button. Active-high reset button. RC oscillator. None of the answers.
The figure shows a:	 Source: http://www.ece.utep.edu	<ul style="list-style-type: none"> Debounce circuit. NAND-based oscillator. Pull-up network Low-pass input filter.
The figure shows a:	 Source: Texas Instruments	<ul style="list-style-type: none"> Transceiver Ring-oscillator A 74HC04 Inverter chip A OEM Standard Advanced Block.
The LED of the figure:	 Source: http://www.seattlerobotics.org/	<ul style="list-style-type: none"> Is ON if the input of the INV is 1. Is ON if the input of the INV is 0. Is utilized as a diode to prevent reverse currents. None of the above answers.

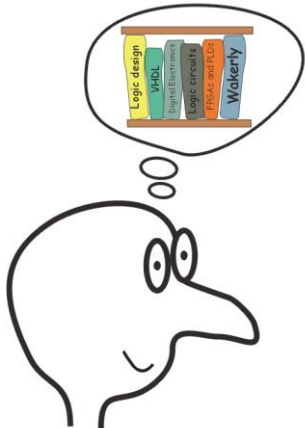
In the table	<table> <tr> <th></th><th>TTL 3.3</th><th>CMOS 2.5</th></tr> <tr><td>VOH</td><td>2.4V</td><td>2.3V</td></tr> <tr><td>VIH</td><td>2V</td><td>1.7V</td></tr> <tr><td>VOL</td><td>0.4V</td><td>0.2V</td></tr> <tr><td>VIL</td><td>0.8V</td><td>0.7V</td></tr> <tr><td>IOH</td><td>-0.4mA</td><td>-4mA</td></tr> <tr><td>IIH</td><td>40μA</td><td>1μA</td></tr> <tr><td>IOL</td><td>16mA</td><td>4mA</td></tr> <tr><td>IIL</td><td>-1.6mA</td><td>-1μA</td></tr> </table>		TTL 3.3	CMOS 2.5	VOH	2.4V	2.3V	VIH	2V	1.7V	VOL	0.4V	0.2V	VIL	0.8V	0.7V	IOH	-0.4mA	-4mA	IIH	40μA	1μA	IOL	16mA	4mA	IIL	-1.6mA	-1μA	<ul style="list-style-type: none"> • IOH is negative because the current comes out of the chip. • IOH is negative because of the current enters to the chip. • IOH is the input current for the logic level high. • IOH is both input and output current for the logic level high.
	TTL 3.3	CMOS 2.5																											
VOH	2.4V	2.3V																											
VIH	2V	1.7V																											
VOL	0.4V	0.2V																											
VIL	0.8V	0.7V																											
IOH	-0.4mA	-4mA																											
IIH	40μA	1μA																											
IOL	16mA	4mA																											
IIL	-1.6mA	-1μA																											
In the table	<table> <tr> <th></th><th>TTL 3.3</th><th>CMOS 2.5</th></tr> <tr><td>VOH</td><td>2.4V</td><td>2.3V</td></tr> <tr><td>VIH</td><td>2V</td><td>1.7V</td></tr> <tr><td>VOL</td><td>0.4V</td><td>0.2V</td></tr> <tr><td>VIL</td><td>0.8V</td><td>0.7V</td></tr> <tr><td>IOH</td><td>-0.4mA</td><td>-4mA</td></tr> <tr><td>IIH</td><td>40μA</td><td>1μA</td></tr> <tr><td>IOL</td><td>16mA</td><td>4mA</td></tr> <tr><td>IIL</td><td>-1.6mA</td><td>-1μA</td></tr> </table>		TTL 3.3	CMOS 2.5	VOH	2.4V	2.3V	VIH	2V	1.7V	VOL	0.4V	0.2V	VIL	0.8V	0.7V	IOH	-0.4mA	-4mA	IIH	40μA	1μA	IOL	16mA	4mA	IIL	-1.6mA	-1μA	<ul style="list-style-type: none"> • VOH is the minimum output voltage for a logic level high. • VOH is the maximum output voltage for a logic level high. • VIH is the minimum output voltage for a logic level low. • VIH is the maximum input voltage for a logic level high.
	TTL 3.3	CMOS 2.5																											
VOH	2.4V	2.3V																											
VIH	2V	1.7V																											
VOL	0.4V	0.2V																											
VIL	0.8V	0.7V																											
IOH	-0.4mA	-4mA																											
IIH	40μA	1μA																											
IOL	16mA	4mA																											
IIL	-1.6mA	-1μA																											
In the table the noise margin is:	<table> <tr> <th></th><th>TTL 3.3</th><th>CMOS 2.5</th></tr> <tr><td>VOH</td><td>2.4V</td><td>2.3V</td></tr> <tr><td>VIH</td><td>2V</td><td>1.7V</td></tr> <tr><td>VOL</td><td>0.4V</td><td>0.2V</td></tr> <tr><td>VIL</td><td>0.8V</td><td>0.7V</td></tr> <tr><td>IOH</td><td>-0.4mA</td><td>-4mA</td></tr> <tr><td>IIH</td><td>40μA</td><td>1μA</td></tr> <tr><td>IOL</td><td>16mA</td><td>4mA</td></tr> <tr><td>IIL</td><td>-1.6mA</td><td>-1μA</td></tr> </table>		TTL 3.3	CMOS 2.5	VOH	2.4V	2.3V	VIH	2V	1.7V	VOL	0.4V	0.2V	VIL	0.8V	0.7V	IOH	-0.4mA	-4mA	IIH	40μA	1μA	IOL	16mA	4mA	IIL	-1.6mA	-1μA	<ul style="list-style-type: none"> • VOH – VIH. • VIH – VOH. • VOH – VOL. • VIH – VIL.
	TTL 3.3	CMOS 2.5																											
VOH	2.4V	2.3V																											
VIH	2V	1.7V																											
VOL	0.4V	0.2V																											
VIL	0.8V	0.7V																											
IOH	-0.4mA	-4mA																											
IIH	40μA	1μA																											
IOL	16mA	4mA																											
IIL	-1.6mA	-1μA																											
Write the right sentence In the Internet Era...		<ul style="list-style-type: none"> • You should not forget to read printed textbooks. • You can search the answers of this App in Google. • You can replace a good professor for Youtube. • You can study and read Whatsapp at the same time 																											

Tabla 4. Preguntas test